Spring 2004

Mobile agent based distributed network management : modeling, methodologies and applications

Jian Ye
New Jersey Institute of Technology

ABSTRACT

MOBILE AGENT BASED DISTRIBUTED NETWORK MANAGEMENT - MODELING, METHODOLOGIES AND APPLICATIONS

by
Jian Ye

The explosive growth of the Internet and the continued dramatic increase for all wireless services are fueling the demand for increased capacity, data rates, support of multimedia services, and support for different Quality of Services (QoS) requirements for different classes of services. Furthermore future communication networks will be strongly characterized by heterogeneity. In order to meet the objectives of instant adaptability to the users' requirements and of interoperability and seamless operation within the heterogeneous networking environments, flexibility in terms of network and resource management will be a key design issue. The new emerging technology of mobile agent (MA) has arisen in the distributed programming field as a potential flexible way of managing resources of a distributed system, and is a challenging opportunity for delivering more flexible services and dealing with network programmability.

This dissertation mainly focuses on: a) the design of models that provide a generic framework for the evaluation and analysis of the performance and tradeoffs of the mobile agent management paradigm; b) the development of MA based resource and network management applications. First, in order to demonstrate the use and benefits of the mobile agent based management paradigm in the network and resource management process, a commercial application of a multioperator network is introduced, and the use of agents to provide the underlying framework and structure for its implementation and deployment is investigated. Then, a general analytical model and framework for the evaluation of various network management paradigms is introduced and discussed. It is also illustrated how the developed analytical framework can be used to quantitatively evaluate the performances and tradeoffs in the various computing paradigms. Furthermore, the design

tradeoffs for choosing the MA based management paradigm to develop a flexible resource management scheme in wireless networks is discussed and evaluated. The integration of an advanced bandwidth reservation mechanism with a bandwidth reconfiguration based call admission control strategy is also proposed. A framework based on the technology of mobile agents, is introduced for the efficient implementation of the proposed integrated resource and QoS management, while the achievable performance of the overall proposed management scheme is evaluated via modeling and simulation. Finally the use of a distributed cooperative scheme among the mobile agents that can be applied in the future wireless networks is proposed and demonstrated, to improve the energy consumption for the routine management processes of mobile terminals, by adopting the peer-to-peer communication concept of wireless ad-hoc networks. The performance evaluation process and the corresponding numerical results demonstrate the significant system energy savings, while several design issues and tradeoffs of the proposed scheme, such as the fairness of the mobile agents involved in the management activity, are discussed and evaluated.

MOBILE AGENT BASED DISTRIBUTED NETWORK MANAGEMENT - MODELING, METHODOLOGIES AND APPLICATIONS

by
Jian Ye

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Engineering

Department of Electrical and Computer Engineering, NJIT

May 2004

APPROVAL PAGE

MOBILE AGENT BASED DISTRIBUTED NETWORK MANAGEMENT - MODELING, METHODOLOGIES AND APPLICATIONS

Jian Ye

Dr. Symeon Papavassiliou, Dissertation Advisor — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Edwin Hou, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Sirin Tekinay, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Filippos I. Vokolos, Committee Member — Date
Assistant Professor of Computer Science, Drexel University

Dr. Lev Zakrevski, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Jian Ye

Degree: Doctor of Philosophy in Computer Engineering

Date: May 2004

Undergraduate and Graduate Education:

- Bachelor of Science in Automatic Control,
  University of Electronic Science and Technology, Chengdu, China, 1993
- Master of Science in Automatic Control,
  University of Electronic Science and Technology, Chengdu, China, 1996

Major: Computer Engineering

Journal and Conference Publications:

Jian Ye, Symeon Papavassiliou, Sirin Tekinay,
"An Efficient and Fair Co-operative Approach for Resource Management in Wireless Networks", submitted to *IEEE Journal on Selected Areas in Communications (IEEE JSAC)*, special issue on Mobile Computing and Networking.

Jian Ye and Symeon Papavassiliou,
"An Analytical Framework for the Modeling and Evaluation of the Mobile Agent Based Distributed Network Management Paradigm", to be submitted (under preparation) to the *International Journal of Network Management*.

Jie Yang, Jian Ye, Symeon Papavassiliou and Nirwan Ansari,
"A Flexible and Distributed Architecture For Adaptive End-to-End QoS Provisioning in Next Generation Networks", accepted by *IEEE Journal on Selected Areas in Communications (IEEE JSAC)*, Special Issue on Intelligent Services and Application in Next Generation Networks.

Jie Yang, Jian Ye and Symeon Papavassiliou,
"Enhancing End-to-End QoS Granularity in Diffserv Networks via Service Vector and Explicit Endpoint Admission Control", *IEE Proceedings - Communications*, vol.151:1, pp. 77-81, February 2004.

Jian Ye and Jiongkuan Hou and Symeon Papavassiliou,
"A Comprehensive Resource Management Framework for Next Generation Wireless Networks," *IEEE Transactions on Mobile Computing*, Vol. 1, pp. 249-264, Oct.-Dec. 2002.

Papavassiliou, S. and Puliafito, A. and Tomarchio, O. and Jian Ye,
"Mobile Agent-Based Approach for Efficient Network Management and Resource Allocation: Framework and Applications," *IEEE Journal on Selected Areas in Communications*, Vol. 20, pp. 858-872, May 2002.

Jian Ye and Symeon Papavassiliou,
"Dynamic Market-Driven Allocation of Network Resources Using Genetic Algorithms in a Competitive Electronic Commerce Marketplace," *International journal of Network Management*, Vol. 11, pp. 375-385, Nov.-Dec. 2001.

Jie Yang, Jian Ye, Symeon Papavassiliou, and Nirwan Ansari,
"Decoupling End-to-End QoS Provisioning from Service Provisioning at Routers in the Diffserv Network Model", submitted to *IEEE Globecom 2004*.

Jian Ye and Symeon Papavassiliou,
"On the Performance Analysis and the Tradeoff Evaluation of the Mobile Agent Based Network Management Approach", *In Proc. Conference on Information Sciences and Systems (CISS2004)*, March 2004.

Jian Ye, Jiongkuan Hou and Symeon Papavassiliou,
"Integration of Advanced Reservation and Bandwidth Reconfiguration based Admission Control in Wireless Networks with Multimedia Services," *In Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems - Workshops*, pp. 844-849, May 2003.

Jie Yang, Jian Ye and Symeon Papavassiliou,
"A New Differentiated Service Model Paradigm via Explicit Endpoint Admission Control," *In Proc. IEEE Symposium on Computers and Communications (IEEE ISCC 2003)*, pp. 299-304, July 2003.

Jian Ye, Symeon Papavassiliou, Giuseppe Anastasi, Antonio Puliafito,
"Strategies for Dynamic Management of the QoS of Mobile Users in Wireless Networks through Software Agents," *In Proc. IEEE Symposium on Computers and Communications (ISCC2002)*, pp. 369-374, July 2002.

Jian Ye, Jiongkuan Hou, and Symeon Papavassiliou,
"Mobile Agent based Framework for Mobility Assisted Channel Reservation in Wireless Networks," *In Proc. Conference on Information Sciences and Systems (CISS2002)*, pp. 833-838, March 2002.

Symeon Papavassiliou, Antonio Puliafito, Orazio Tomarchio and Jian Ye,
"Integration of Mobile Agents and Genetic Algorithms for Efficient Dynamic Network

Resource Allocation," *In Proc. IEEE Symposium on Computers and Communications (ISCC2001)*, pp. 456-463, July 2001.

Jian Ye and Symeon Papavassiliou,
"Dynamic Market Driven Provisioning of Network Resources Using Genetic Algorithms in a Competitive Electronic Commerce Marketplace," *In Proc. Conference on Information Sciences and Systems (CISS2001)*, pp. 519-524, March 2001.

To my beloved wife
and
our parents

ACKNOWLEDGMENT

I would like to express my gratitude to my advisor, Dr. Symeon Papavassiliou, for his advice, guidance and support throughout my Ph.D. studies. His encouragement of my work and commitment to whatever research topic we would undertake was inspiring, and has made the time I have spent here enjoyable. His technical and editorial advice was essential to the completion of this dissertation and has taught me innumerable lessons and insights on the workings of academic research in general.

My thanks also go to the members of my dissertation committee, Dr. Edwin Hou, Dr. Sirin Tekinay, Dr. Filippos Vokolos and Dr. Lev Zakrevski for reading previous drafts of this dissertation and providing many valuable comments that improved the presentation and contents of this dissertation.

The friendship of Jiongkuan Hou, Jie Yang and Chengzhou Li is much appreciated and has led to many interesting and good-spirited discussions relating to this research. I am also grateful to my colleagues Jun Jiang and Sheng Xu for their help.

Last, but not least, I would like to thank my wife Fan for her understanding and love during the past few years. Her support and encouragement was in the end what made this dissertation possible. My parents, Ruquan Ye and Meiyan Ai, receive my deepest gratitude and love for their dedication and the many years of support during my school studies that provided the foundation for this work.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter Page

TABLE OF CONTENTS
(Continued)

Chapter Page

TABLE OF CONTENTS
(Continued)

Chapter Page

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Network Management Overview

Currently, network management systems follow a platform-centric manager-agent paradigm [1]. By using certain management protocol such as the Simple Network Management Protocol (SNMP) in data networks or the Common Management Information Protocol (CMIP) in telecommunication networks, raw instrumentation data are collected by agents embedded in network elements (NEs) and are sent to central management platform for further processing. Unlike the centralized network management system that there is only one network manager in the domain (as shown in figure 1.1), current network management systems generally are organized in a *hierarchical* structure, which divides the network into a few subdomains and there is a Management Station (MS) (manager) in each subdomain (figure 1.2). Each subdomain's MS is only responsible for the NEs within its coverage. At the higher level, there is also a manager of managers that controls the submanagers in the subdomains. By placing the manager and submanager at different levels of the hierarchical architecture, this management structure can be expanded to a multi-level management structure. Several hierarchical management applications that range from Quality of Service (QoS) routing to diagnostic algorithms haven been described in the literature [2, 3, 4].

Although the hierarchical management structure can partially solve some of the problems of the centralized management structure, such as low scalability, low reliability etc, it is still not an *inherently distributed management architecture* and does not provide for a flexible architecture to accommodate new features and services as the Internet evolves [5].

Internet protocol (IP), which is a universal network-layer protocol for wireline networks, is also becoming a promising universal network layer over all wireless systems.

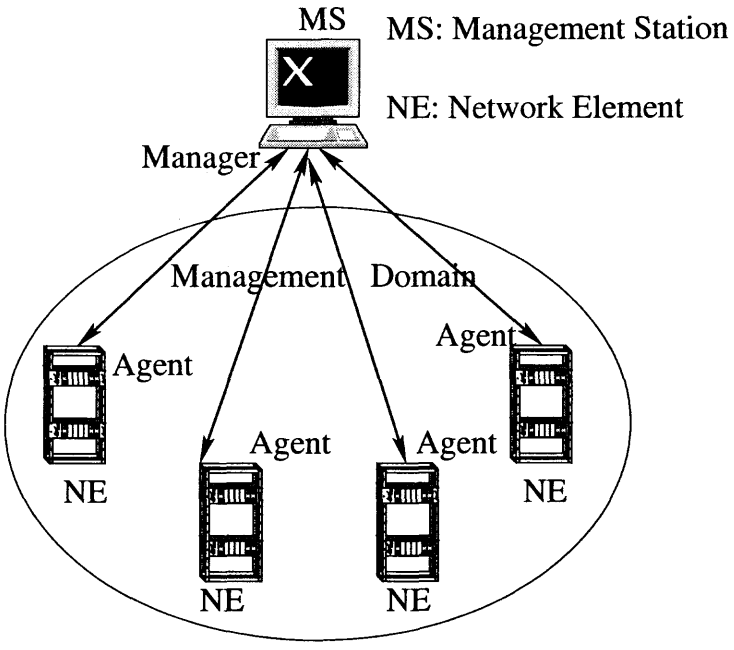


Figure 1.1 Centralized network management framework

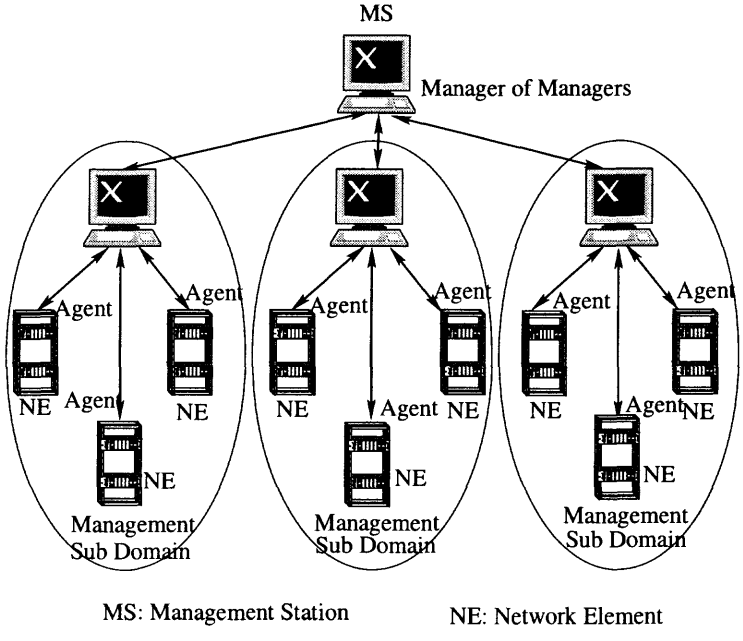


Figure 1.2 Hierarchical network management framework

The explosive growth of the Internet and the continued dramatic increase for all wireless services are fueling the demand for increased capacity, data rates, support of multimedia services, and support for different QoS requirements for different classes of services. Therefore issues associated with the QoS, network management and control, introduction of new applications and system adaptability, are fast gaining critical research and commercial importance.

Furthermore another feature that will strongly characterize the future telecommunication networks is heterogeneity. Various access technologies and standards co-exist in the world today, and the future communication network that will evolve from the current networks is expected to be a combination of various multi-domain, multi-provider, multi-technology wired/wireless systems. For instance, a sample next generation networking topology is shown in figure 1.3. Today's many different wireless systems, ranging from Personal Area Networks (PANs) and Wireless Local Area Networks (WLANs), to wide-area cellular systems, are often not compatible with each other, which makes it difficult for a user to roam (handoff) from one radio system to another. In order to meet the objectives of instant adaptability to the users' requirements, and of interoperability and seamless operation within the heterogeneous 4G environment, flexibility in terms of network and resource management will be a key design issue for the future mobile network architectures.

1.2 Client-Server Based Network Management Mode

The computing paradigm for the traditional network management is based on Client-Server mode. The Client-Server paradigm leaves the network elements with passive roles in the management procedure. The only function of the agent embedded inside the network element is to collect raw data for management purposes. All the computation/decisions are made at the management stations. Specifically, in the Client-Server paradigm, data distributed in different nodes for a management task are collected by a static agent

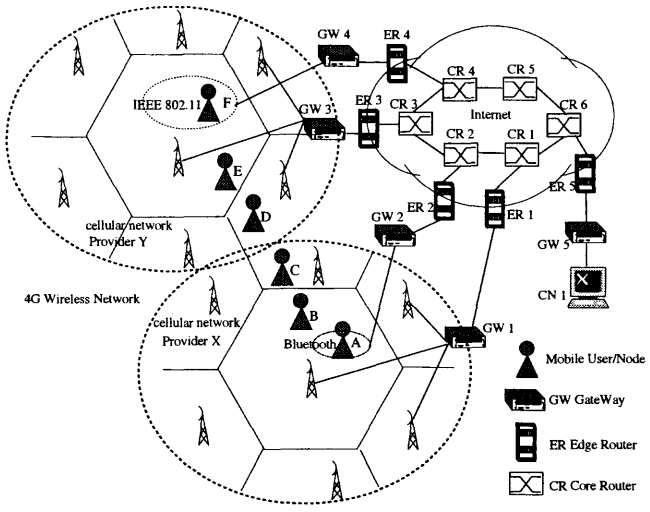


Figure 1.3 Hybrid wired-wireless network

embedded in these nodes. For management purposes, these data may be organized in a Management Information Base (MIB). When some management entity at some node V_0 need data from node V_i ($i \neq 0$), it will send a request to the agent at that node. The agent then gets the data from the MIB and sends it back to the management entity, which in turn applies the algorithm on the data. If the management entity wanted to manipulate the remote device at node V_i (e.g. make configuration changes according to the computation result), it sends a configuration command along with the configuration specification to the node. During this procedure, requests, replies and data are transfered among the nodes involved into the management activity. We call these information exchanges remote interactions.

With the progress of micro-electronic technology, today's network elements are becoming more and more powerful and capable of executing management tasks by themselves. Therefore in principle it is not necessary anymore to restrict the management functions at the management stations. Based on these concepts, **Management by Delegated** (M_bD) has been proposed in [1] - "management processing functions can be delegated dynamically to the network elements and executed locally rather than centrally".

The philosophy behind the M_bD is that, instead of moving raw data (management information) from the NEs to the MS, management algorithms are sent to the NEs by the means of mobile codes or mobile agents. Figure 1.4 demonstrates this difference.

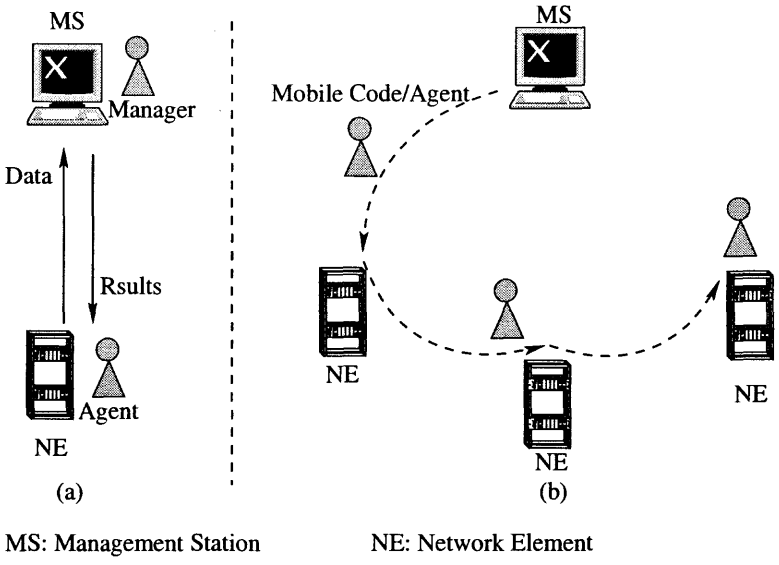


Figure 1.4 (a): Client-Server mode (b) Mobile Code/Agent mode

1.3 Mobile Agent Technology (MAT)

The new emerging technology of agent programming has arisen in the distributed programming field as a potential flexible way of managing resources of a distributed system and is a challenging opportunity for delivering more flexible services and dealing with network programmability. Distributing intelligence across the network allows the fast exploitation of more advanced services that can dynamically adapt to the user's requirements. The most salient feature that distinguishes agents and ordinary code is autonomy. Agents can cooperate with other agents to carry out more complex tasks than they themselves could handle. One special kind of agent, Mobile Agent (MA), may move from one system to another to access remote resources [6]. This technology itself has two basic features: mobility and agency [7]. Mobility means that the entity of the agent can move freely among network elements via proper MA supporting platform, while agency

implies that the MA can move and act among NEs automatically on behalf of some other entity. Mobile agent technology enriches the design of network computing paradigm, including the network management process.

Some of the advantages of using MAs for network management purposes include [8]:

- Reduced network load: Moving the computation (codes) to the data rather than the data to the computation (management station) could reduce the traffic in the network, especially when very large volumes of data are stored at the remote host.

- Reduced network latency (reaction time) which is critical for several real time management tasks (e.g. support of mobile handoff).

- Asynchronous and autonomous execution: This capability is important for large scale networks and management systems that support large numbers of NEs.

- Adaptability: *intelligent* agents could make decision themselves depending on the local information. They could also have the capability to distribute themselves into network elements and cooperate with others for a certain management task.

- Scalability: traffic and computational load can be balanced among nodes involved into the management tasks.

- Reliability: the remote interactions among different nodes may be reduced, and the computation is affected less by the network environment or the central management station.

- Flexibility and support of heterogeneous systems: The operations of mobile agents only depend on the supporting platform and are independent of the NEs. This feature is especially beneficial when the network includes several hybrid systems based on different technologies (hardware and/or software).

1.4 Background and Research Objectives

1.4.1 Motivation and Objectives

Since mobile agent technology emerged as a promising paradigm and is much more flexible and dynamic than the Client-Server paradigm, it has recently become an active research and development topic. Specific problems of interest in this field include: a) MA supporting/management platform design, b) mobile agent system security, c) modeling and analysis of mobile agent based management systems, and d) management applications. The latter two directions are the main focus of this dissertation.

Mobile agent platform design emphasizes the design of a general mobile agent supporting system that has the functionality to host, create, transport, track, execute, suspend, resume, stop and kill mobile agents. The platform also may support communications among groups of agents. Several MA platforms have been reported in the literature: **MARS** [9] at University of Modena and Reggio Emilia, **MAP** [10] at University of Catania , **D'Agents** (formerly **AGENT TCL** [11]) at Dartmouth College, **PLANGENT** [12] at Toshiba Corp., and **Phoenix** [13] at Intel Corp.

Security is another crucial aspect of agent systems. An adequate security model is required in order to assure a high level of protection to the agents and the nodes, so that nodes can be protected from the attacks of unsafe agents, and the agents can run on the nodes of the network without being damaged. Although no security standard has yet been set for mobile agent systems, and the issues associated with the agent security is still an open active research topic, several solutions have been proposed in the literature trying to solve the related problems [14, 15, 16].

Several activities that utilize MA based technology for specific traditional network management functions have been reported in the literature. For instance in [17] an MA based system is proposed that can help individuals easily access various networking resources in systems that integrate realtime voice and non-realtime messaging services. In [18] MA technology is used to enhance telecommunication services engineering, while

[19] employs MA for QoS routing. Other applications include performance monitoring [20], QoS management [21] and distributed measurements [22].

In most of the above mentioned effort the emphasis is still placed on the platform used to support the specific management function, while qualitative arguments are mainly made regarding the benefits of the use of the agent technology. Very little work has been done in the area of modeling and analysis of the mobile agent based paradigm in order to provide a framework that would allow a more in-depth and quantitative evaluation of the benefits and tradeoffs of the MA paradigm vs. the traditional CS paradigm. In [23], the authors compare the performance of mobile agent based computing paradigm with the Client-Server paradigm in the scenario of data filtering application within wireless networks. Assuming a pipeline model of how the filtered data is passed from Internet server to the wireless mobile user, tradeoffs of the computational capacity of the Internet server or mobile clients and latency of a task are discussed. In [24] a stochastic model is proposed, that can be used to describe some behavior such as dwell time, life span and reporting process of mobile agents, when they are used in network management systems. However, since some statistical parameters used in that model are very difficult to be identified , this approach is hard to be used as a guideline in practice. In [25] and [26], four main paradigms for network computing are identified: Client-Server, code-on-demand, remote evaluation and mobile agents. Analytical models of the overall traffic around the management stations for the data retrieving application are proposed. From the system's point of view, remote evaluation and code-on-demand are symmetric paradigms and could be classified as extreme weak mobile agent paradigms: only the algorithm code is transfered between two nodes instead of a computing entity (mobile agent). In [27] the authors attempt to build a more comprehensive model by including additional factors, such as traffic management cost, security overhead, etc., in the overall system modeling, for network monitoring and data searching applications. However, those analytical models proposed in the literature

are tightly coupled with the specific application under consideration (e.g. data searching, data filtering) and can not be easily extended to other applications.

These observations and challenges have motivated our research concerning the following two interrelated areas: a) the investigation and development of models that would provide a generic framework for the evaluation and analysis of the performance and tradeoffs of the corresponding mobile agent management paradigm; b) the application of mobile agent technology in both wired and wireless networks for resource allocation and management purposes.

1.4.2 Contributions

The main contributions of this dissertation are summarized as follows:

- The use and the benefits of the mobile agent based management paradigm in the network and resource management process is demonstrated, through the introduction of a commercial application of a multioperator network, and the investigation of the application of agents to provide the underlying framework and structure for its implementation and deployment. It is demonstrated that the mobile agent technology helps overcome several implementation and design issues associated with the need for distributed operation of various network and resource management tasks and algorithms. Such considerations include balancing of the resource management computational load in the network, dynamic fault tolerance, customized service provisioning.

- A general analytical model and framework for the evaluation of various network management paradigms is introduced and discussed. It is also illustrated how the developed analytical framework can be used to quantitatively evaluate the performances and tradeoffs in the various computing paradigms, by comparing the performances of the mobile agent based paradigm with the corresponding ones of the Client-Server mode, under different scenarios.

- The design tradeoffs for choosing the MA based management paradigm to develop a flexible resource management scheme in wireless networks is discussed and evaluated. The integration of an advanced bandwidth reservation mechanism which facilitates the efficient and seamless operation of the handoff process, with a bandwidth reconfiguration based call admission control strategy that supports flexible QoS management, is also proposed. A framework based on the technology of mobile agents is introduced for the efficient implementation of the proposed integrated resource and QoS management.

- The use of a distributed cooperative scheme among the mobile agents is introduced, that can be applied in the future wireless networks, to improve the energy consumption for the routine management processes of mobile terminals, by adopting the peer-to-peer communication concept of wireless ad-hoc networks. The performance evaluation process and the corresponding numerical results demonstrate the significant system energy savings that can be achieved through the use of the mobile agent cooperative approach, while several design issues and tradeoffs involved in the proposed scheme, such as the fairness of the mobile agents involved in the management activity are identified and analyzed.

1.4.3 Organization

The remaining of this dissertation is organized as follows.

In chapter 2, a commercial application of a multioperator network is introduced, and the use of agents to provide the underlying framework and structure for its implementation and deployment is demonstrated. This chapter mainly demonstrates the use and the benefits of the mobile agent based management paradigm in the network and resource management process, and serves as the motivation for the work presented in the following chapter.

In chapter 3, a general analytical model and framework for the evaluation of various network management paradigms is introduced and discussed. First, a generic interaction

model is introduced to provide a representation and description of the activities involved in network management applications. Based on this generic model, various important performance metrics are introduced and calculated analytically. Then we illustrate how the developed analytical framework can be used to quantitatively evaluate the performances and tradeoffs in the various computing paradigms, by comparing the performances of the mobile agent based paradigm with the corresponding ones under the Client-Server mode under different scenarios.

In chapter 4, the integration of an advanced bandwidth reservation mechanism which facilitates the efficient and seamless operation of the handoff process, with a bandwidth reconfiguration based call admission control strategy that supports flexible QoS management, is proposed. A framework based on the technology of mobile agents is introduced for the efficient implementation of the proposed integrated resource and QoS management. Using the model and analytical evaluation process discussed in chapter 3, the design tradeoffs for choosing the MA based management paradigm are discussed, while the achievable performance of the overall proposed management scheme is evaluated via modeling and simulation.

In chapter 5, a cooperation scheme among some of the mobile agents of the management framework proposed in chapter 4 is discussed in detail, to enhance the performance of a mobile agent based management system. Specifically we propose and demonstrate the use of a distributed cooperative scheme among the mobile agents, that can be applied in the future wireless networks to improve the energy consumption for the routine management processes of mobile terminals, by adopting the peer-to-peer communication concept of wireless ad-hoc networks. The proposed scheme consists mainly of two parts: the distributed clustering algorithm and the query-reply-report information delivery interactions among the cluster head, member terminals and the base stations. The performance evaluation process and the corresponding numerical results presented demonstrate the significant system energy savings that can be achieved through

the use of the cooperative approach. Furthermore, several design issues and tradeoffs of parameters involved in the proposed scheme, such as the fairness, the cluster size, the query range, etc. are discussed, analyzed and evaluated.

Finally, chapter 6 concludes the dissertation by summarizing the main contributions and conclusions of this work, and discussing the directions for future work.

CHAPTER 2

USING MA IN MARKET DRIVEN NETWORK PROVISIONING

In this chapter we demonstrate the use and benefits of the agent based approach, by presenting a commercial application of a multioperator network, where the use of mobile agents can provide the underlying framework and structure for its implementation and deployment. Specifically, we describe the efficient integration and adoption of mobile agents and genetic algorithms in the implementation of a valuable strategy for the development of effective market based routes for brokering purposes. In section 2.1 we first introduce the reference multioperator network model, and formulate the routing problem under consideration. A genetic algorithm is proposed in section 2.2, which provides a kind of stochastic algorithm searching process in order to identify efficient resource allocation and routing strategies. Because of the distributed nature of the routing problem, the GA routing algorithm is integrated into the mobile agent based management architecture to facilitate the implementation of this approach in an efficient way. The integration of the Genetic Algorithm (GA) and mobile agent management paradigm is discussed in section 2.3. Finally, in section 2.4, we present some numerical results that demonstrate the operation and effectiveness of our strategy in some test cases, while section 2.5 concludes this chapter.

2.1 Multi-Operator Network Model and Problem Formulation

In the wide area electronic commerce communication services and applications in an open marketplace many types of providers are usually involved in order to complete the end-to-end service offering. Specifically, the Service Provider (SP) is responsible for the definition of the service characteristics and the maintenance of the customer premises equipment, while the Network Provider (NP) provides the network infrastructure (i.e. high-speed network). The Network Provider relieves the other parties involved in that arrangement of the cost and effort of network management by reducing labor cost and

capital investment. In such an arrangement, the Service Provider is essentially a customer of the Network Provider, while the Service Provider provides the service to its own customers or end-users (usually multiple customers with small to medium size). As a means of competition many different Network Providers offer access to a remote Content Provider (CP), which provides multimedia services such as voice, data and video. A Service Provider is capable of accessing many different Network Providers to request service. The various network operators (providers) will then be competing to sell their network links to clients through a representative agent host, the Network Access Broker (NAB). A SP may have more than one attachment points with a NP, for load distribution, redundancy and reliability purposes. The Service Provider (SP) acts as a single point of contact between a Client and multiple NABs, rather than the client having their own negotiating agents hoping between multiple network hosts. Hence several different customers' resource requirements can be aggregated in order to achieve a lower cost to the individual.

For the sake of representation simplicity we can assume that for each network there is a Management Station (MS), which maintains routing tables, network pricing, resource updates, and other management information and network statistics. The Inter Network Access Broker (INAB) serves as an intermediary between two neighboring networks. This allows negotiations for access to one another's resources. This is useful either when one network is unable to supply the requested end-to-end bandwidth asked by a SP agent, or when the SP determines that the combination of parts of two or more network providers could provide a more efficient and/or cost effective (for the user) resource selection. Overall such an approach tends to offer to the user the best available resource allocation and cost at the time of the request, and also allows network providers to avoid losing potential revenue (by turning down a connection) if they are unable to provide a complete end-to-end cost efficient connection at the time of the request. Rather than turning down the request, Network A can complete portion of the connection within its network while supplement

bandwidth will be provided from Network B, to compensate for any shortfalls or the inefficient solutions.

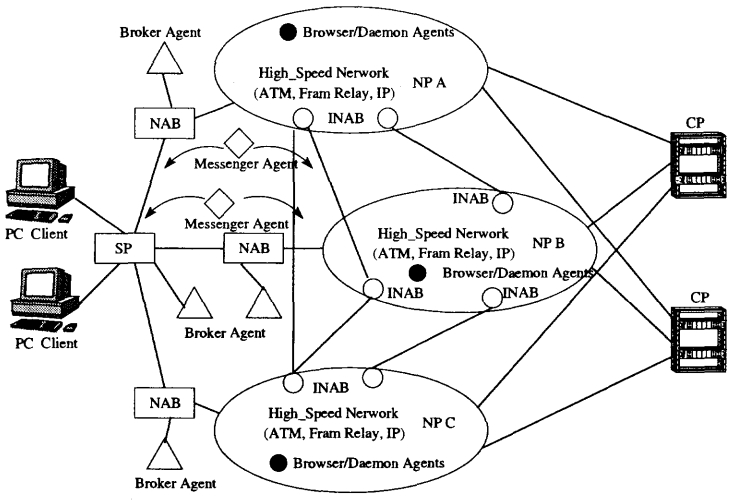


Figure 2.1 Network model for agent brokering environment

A scenario that involves three competing network providers and two service providers is illustrated in figure 2.1. The three networks are owned by three different operators, and each one of them is responsible for resource allocation and pricing strategies within its own environment. Moreover the three networks have some interconnection points with each other therefore allowing traffic to flow among the different networks, as it is expected in an open marketplace. The SP is informed periodically about the cost changes associated with each of those interconnection points. As a means of competition all three networks offer the same access to the remote CP.

The SP serving a particular client is responsible to identify the best/optimal connection path per request of the customer. Such a path may traverse one only network or could traverse multiple networks. In the first case the only responsibility of the SP is to select the best route at the time of the request, as reported by each individual network provider. In the second case the SP is responsible for identifying the optimal solution that could result as a combination of all the possible route scenarios via multiple network providers. In this case the identification of the optimal solution becomes a complex

optimization problem since it involves many parameters and the set of feasible solutions can become very large.

In general the identification of optimal routes in such an environment becomes a complex optimization problem since it involves many parameters and the set of feasible solutions can become very large. In our research work, a genetic algorithm is used as a means to solve this problem. In general, QoS routing with multi-constraint has been shown to be NP-hard problem in the literature (e.g. [28]). At the same time GA presents a good method to handle multi-constraint optimization problems and does not depend on the properties of the problem itself. Due to the intrinsically distributed nature of the problem under consideration, the combination of GA's parallel property that provides an efficient method for searching large spaces can provide an overall efficient and successful solution to the routing and resource management problems.

According to the changes in the different network conditions, Service Provider (SP) always wants to find a cost-reasonable route for its costumers. We assume that at certain time period, SP knows: traffic from SP to CP; QoS requirements/constraints (i.e. time delay constraint from SP to CP); connectivity of nodes of Network Providers(NPs); bandwidth available for each link between nodes; cost for the traffic to pass through each link and time delay for the traffic to pass through each link.

As the number of nodes in the network increases, the searching space of finding a route increases dramatically. Therefore it is not practical for an SP to find an optimal route all the time. Among the objectives of this chapter is to develop a practical approach of finding a cost-efficient route from SP to CP via nodes of multiple NPs according to the state of the networks at that time. It should be noted here that the nodes of interest represent the access nodes from SP to the individual networks, the edge nodes that provide interconnectivity among the various Network Providers (i.e. INABs), and/or egress nodes providing connectivity to the CPs. According to the multi-operator network model described in previous section, each network provider is responsible for the resource

management, routing updates and route discovery process within its own environment (i.e. only among nodes belonging to this network). Any changes within the context of a single operator are reflected by potential changes in cost and delays of the links (paths) connecting the access nodes, edge nodes and/or egress nodes. Our algorithm aims to identify the "best route" that could result as a combination of all the possible route scenarios via multiple network providers.

In the following, we provide the problem formulation, under some model simplifications and assumptions regarding the constraints. Specifically the connectivity constraints and bandwidth requirements are handled as follows: for those pairs of nodes that either do not have physical link between them, or the link presents insufficient available bandwidth to handle a request, we assume that the corresponding link has a very large delay. Thus, the problem can be described as the following optimization problem.

$$Min \quad Cost(route) \tag{2.1}$$

$$s.t. \quad Delay(route) \leq MaxDelay \tag{2.2}$$

2.2 Genetic Algorithm Description and Operation

In general Genetic Algorithm (GA) is kind of stochastic algorithm searching process in the solution space by emulating biological selection and reproduction. Indeed, GA has been used to solve various optimization problems [29] and in many cases have obtained very good results.GA only uses values of the objective function for optimization to select genes. This makes GA very attractive as an optimization tool since it doesn't need to know the detailed information of the system. In general we can say that GA is robustly applied to problems with any kind of objective function. Because of GA's parallel property, it is very suitable for large searching space cases. In this section we propose a genetic-based algorithm in order to address the problem described in the previous section. In the following we describe the different features and phases of our algorithm.

2.2.1 Encoding Approach

Generally, GA can not operate directly in the search space of a specific problem. Therefore, we must map the search space of the specific problem into the space where GA can operate. This process is called encoding of GA. In the literature related work in using GA as an optimization tool for finding routes for vehicles [30] or using GA to solve the famous Traveling Salesman Problem [31] has been reported. There are several encoding approaches mentioned in literature, such as Adjacency Representation, Ordinal Representation, and Path Representation. Some encoding approaches are not suitable for GA operation(crossover and mutation), and some other encoding approaches are inefficient in searching the solution space. In our work we choose path representation approach to naturally encode a route due to its easy implementation. That is, we list all the nodes that the route passes through in sequence. In order to encode the route in a fixed data structure, we fill "0s" into the empty space of the code.

For instance, in a scenario with 10 nodes in addition to the CP and SP, we can use an array with 10 elements to represent the route. Once the number of nodes that the route passes is less than 10, the corresponding element will be zero. In this case, we will use [1 2 3 4 0 0 0 0 0 0] to represent the route bellow: SP–1–2–3–4–CP.

2.2.2 Population Initialization

In order to start GA computation, we need to identify some initial population. In population initialization process, we can randomly determine how many nodes the route will pass through and randomly determine which node will be in the route and the sequence of the nodes of the route. However, there will be some solutions that may violate constraints of delay or inter-connectivity. We use "penalty method" to deal with these constraints, as follows: a) For those links that do not exist, we assign a very large delay value to them, b) For those routes that violate the delay constraint, we add a penalty to their cost. In our algorithm, we use the following expression (2.3) to evaluate the weighted cost of those

illegal routes.

$$Cost'(route) = Cost(route) * (\alpha + \frac{Delay(route)}{MaxDelay}) \tag{2.3}$$

Where, $Cost'(route)$ is the weighted cost of route, $Cost(route)$ is the function that evaluates the total cost of the links that the route may pass through; $Delay(route)$ is the function that evaluates the time delay of the route; MaxDelay is the upper-bound of the time delay constraint, α is the penalty constant(in our algorithm it is set to 2).

2.2.3 Fitness Evaluation

Fitness of the solutions are proportional to their survivability during GA computation. Selection operation mentioned in next subsection will use these values to keep "good" solutions and discard the "bad" solutions. For simplicity, we can use the cost of each route as the fitness of each solution. However, in order to prevent those solutions with very low fitness being discarded by Selection Operation of GA at the first several computation loops of GA (this is called premature of GA), we enlarge the value of fitness of those "bad" solutions. In this way, some "bad" solutions still have chances to survive at the beginning of the evolution of GA and the "good" parts in them will have chances to transfer to new generations. For convenience, we normalize the fitness of solutions to [0,1] by the following expression(2.4).

$$Fitness(route) = \frac{(Maxcost - Cost(route)) * Mincost}{(MaxCost - Mincost) * Cost(route)} \tag{2.4}$$

Where MaxCost and MinCost are maximum and minimum cost of routes in each generation of population, respectively. In first 30 computation steps of GA, we adjust those fitness less than 0.005 to 0.005 and this helps to prevent premature efficiently.

2.2.4 Selection Operation

The purpose of Selection Operation in GA is to keep "good" solutions while discard "bad" solutions at the same time. So after selection operation, those solutions that have high fitness value will survive while those solutions with low fitness value will be discarded. This operation tries to simulate "Natural Selection" in real life.

Two selection operators are used in our algorithm. The first one is based on the "Fitness proportional model". It is also called "Monte Carlo selection". The algorithm is as follows:

(1) Add the fitness of all solutions; (2) Randomly generate a number between 0 and the sum of fitness; this number is called pointer of Monte Carlo wheel; (3) Add fitness of each solution one by one until the value is greater than the pointer. Then the last solution is being selected.

Using this algorithm, the higher the fitness value, the bigger the chance of that solution being chosen. The second selection operator used in our algorithm is the "best solution Reservation". The best solution in the population will always survive and several duplicated copies will be generated for mutation operation. In this way, the GA will always converge to a certain "good" solution. Moreover, there will be good chance to find better solution on the base of the best solution of each generation.

2.2.5 Crossover Operation

Because of Crossover Operation, any pair of solutions in the population have chance to exchange part of their solution information with others. Therefore those "good" parts from different solutions may be combined together to create a new, better solution. The two original solutions are the "parents" of the new solutions generated.

There are are many crossover operators designed to solve different problems. K. Uchimura et al. ([30]) proposed a new kind of adjacency based operator in Vehicle Routing Problem. However this operator is problem specific operator and uses complicated

computation in order to obtain good results. In this chapter, we use traditional one-point crossover method. That is, we find a certain point of the array and swap the part before and after the cross-point to generate two new solutions. However, because the number of nodes the route may pass through is not fixed, it is difficult to determine a fixed crossover point. So we designed a new dynamic crossover point method. The crossover point is determined by: $[(A + B)/4]$, where A and B are numbers of nodes two routes will pass through respectively, and the operator "[ ]" is a rounding function. For example, let us assume that before crossover operation, there are two routes: [1 2 3 4 5 0 0 0 0 0] and [6 7 8 0 0 0 0 0 0 0]. According to this procedure ($[(A + B)/4] = 2$) after crossover, we get: [1 2 8 0 0 0 0 0 0 0] and [6 7 3 4 5 0 0 0 0 0]. Note that, only part of the population will experience crossover operation; this rate is called crossover rate.

2.2.6 Mutation Operation

In mutation operation, we randomly choose a solution in the population and then change the solution slightly to generate a new solution. In this way, we have chances to find better solution that can not be found by only crossover operation. In our algorithm, four mutation operators have been designed:

(1) Randomly delete a node from a route; (2) Randomly add a node into a route; (3) Randomly delete two nodes from a route; (4) Randomly add two nodes into a route.

Only part of population will experience the mutation operation (this is called mutation rate). In order to enhance the local searching ability, those copies of best solution of each generation are all treated by mutation operator. In this way, GA may find out better solution that is "close" to the best solution of each generation.

2.2.7 Repair Operation

During crossover and mutation operation, illegal representation of route may be generated because duplicated elements (nodes) may appear in the same route. In our algorithm, we

delete those duplicated nodes that will bring high cost to the route. For example, there may be a route like [1 2 3 4 5 3 7 0 0 0], where node "3" is duplicated. We can evaluate cost of strings (2 3 4), (5 3 7) and delay of strings (2 3 4), (5 3 7). If the weighted cost of string (2 3 4) is less than (5 3 7), we will discard node 3 in (5 3 7).

2.2.8 Finding Route Efficiently and Dynamically

Though GA can be used to search in large solution space and obtain an optimal solution, it usually takes a lot of time to converge to the optimal solution or GA even can only find some other suboptimal solution instead. In practice, we usually want to find a suboptimal solution which however is close to the optimal one. So in this approach, we use a combination of conditions to determine when to stop our algorithm's computation. The basic idea is as follows. After a minimum number of trails(MinTrails), if the algorithm has found feasible solution and has made no more improvement for a specific period of time, we will have it stopped. In our algorithm the "improvement" is presented by the average cost change rate of the best solution of certain generation. This change rate is evaluated by the following expression(2.5).

$$ChangeRate(k) = \frac{(cost(i) - cost(i+1))}{(k-i)} \tag{2.5}$$

where, we assume the cost of the best solution of that generation changes at ith step and $ChangeRate(k)$ is the average change rate of cost at kth step($k > i$). Once this value is less than a certain lower-bound(MinChangeRate), we may stop GA computation.

As time passes by, the price of each link and congestion of the network will change gradually. So when new traffic comes, SP will re-compute routes for customers. During the dynamic operation of the system, in oder to improve the efficiency of our algorithm, we considered the approach of taking advantage of results of last computation. Surat Tanterdtid etc. [32] has discussed how to re-use past solution for adaptive VPs assignment. They proposed to mix certain "training genes" into the initial population of the new route

computation. However, we found that this will lead to premature and prevent GA to find better solutions. Instead of mixing past solution into initial solution of GA, we mix the past solution into population after 70% of MinTrails of GA loops. In this way, we can still take advantage of the results of last computation and prevent premature at the same time. If the network conditions change smoothly, we can take advantage of the past best solution of last computation. If the network conditions change dramatically at a certain time and the optimal route may totally differ from the past solution, GA will not take advantage of past best solution by mixing past solution into the population during GA computation.

2.2.9 Flow Chart of the Algorithm

The following flow chart (figure 2.2) summarizes the algorithm's operation.

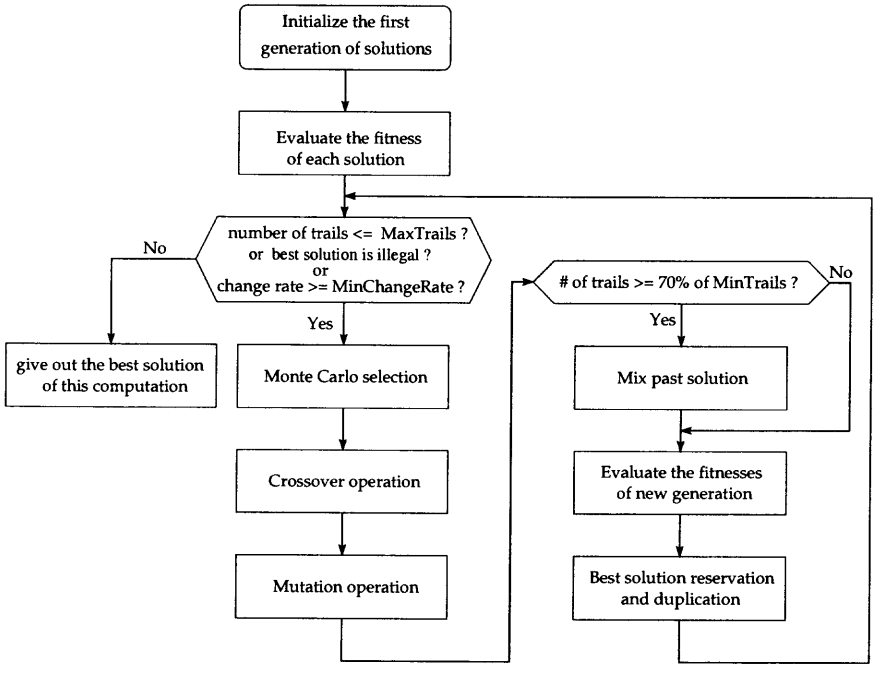


Figure 2.2 Flow chart of GA

2.3 Integration of Genetic Algorithms and Mobile Agents

Due to the intrinsically distributed nature of the optimization problem under consideration, mobile agent based management architecture can be successfully adopted to collect the input data of the genetic algorithm. A set of daemon and messenger agents will monitor and retrieve traffic measures from SP to CP, while other daemon agents will monitor and evaluate the time delay for the traffic to pass through each link. Due to the flexibility of the agent system, the measure of a new and different parameters can be added to the system in a very efficient way. In fact, it is necessary only to develop the agent with the actual code for getting that parameter: then it can be dynamically executed in the most convenient place of the network.

What is important to stress here is how the underlying agent architecture may interact with the genetic algorithm itself. We assume that an SP may host a particular kind of agent (named *broker agent*) which is in charge of identifying the optimal path to manage a specific connection request. The interaction among a broker agent and the algorithm may occur according to the following strategies.

1. The broker agent is able to execute the algorithm in run-time upon the request from the PC client.

2. The broker agent sends a request to a network node where the genetic algorithm can be executed. The optimal path is then sent back to the broker agent, which activates the setup procedure.

3. A set of optimal paths for different pairs of PC client and content providers are stored in a database (eventually distributed) which is accessed from the broker agent to retrieve the more convenient path to satisfy the specific request. Once the connection is established, the genetic algorithm can be re-executed in order to identify a more convenient path, if the case.

These various strategies present many different trade offs that range from the distributed nature of the strategy and its corresponding complexity, to the accuracy of the network state information and optimality of the identified routes, to the users' and providers' specifications and requirements. For instance the third strategy does not assume a real-time execution of the genetic algorithm, thus allowing a fast satisfaction of the user request. Such choice might not be optimal; this is why a new path could be identified by an off-line execution of the algorithm. Such an approach also allows a loose interaction with agents dedicated to the retrieval of fresh and updated parameters to be used as input to the genetic algorithm. Of course, if network performance and reliability change due to some reason, monitoring agents distributed in the system will promptly react to pass the genetic algorithm the new data to recompute the new optimal paths.

In the following we provide a detailed description of the actual integration of the genetic based algorithm in the framework of mobile agents and identify the roles of the individual elements (e.g. various agents) involved in the overall operation. Specifically, as shown in figure 2.3, Broker Agents (BrkA), Messenger Agents (MA), Browser Agents (BA) and Daemon Agents (DA) are used to migrate among different network elements to implement the proposed routing algorithm. Even though GA is proposed to solve the optimization problem in our reference model, other method could also be used and can be embedded into our MA based framework.

Once the PC client needs a connection to the CP, a Messenger Agent will be sent from PC client to SP containing information about the upper bound of setup time delay of the connection and the corresponding QoS requirements. After receiving the Messenger Agents from PC client, SP creates a Broker Agent to deal with this connection requirement. This Broker Agent creates Messenger Agents containing source and destination information, as well as QoS requirements, and multicasts the agents to each NAB that it is connected with. Then the Broker Agent in SP waits for the Agents from NABs to obtain the routing solution according to the scheme depicted in figure 2.4.

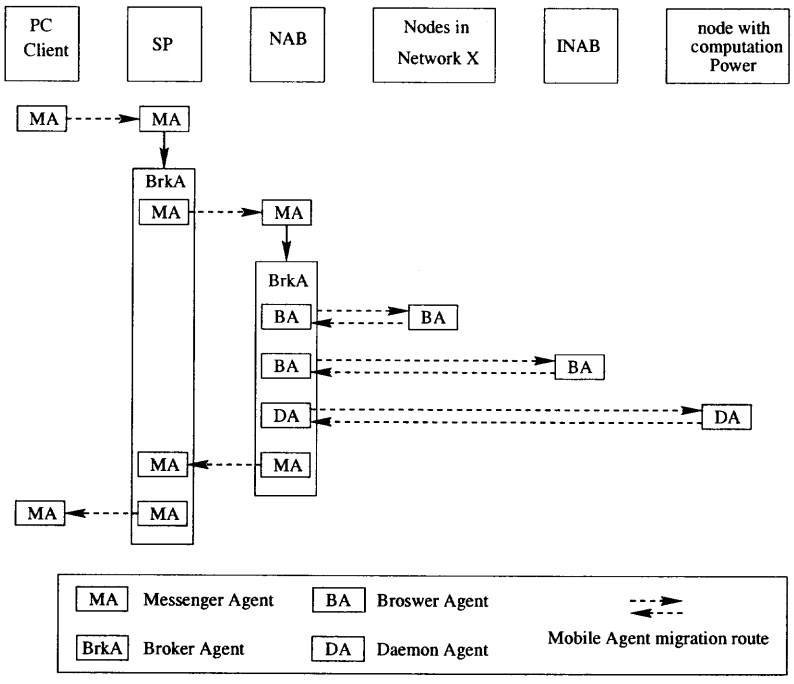


Figure 2.3 Agents used to implement routing algorithms based on GA

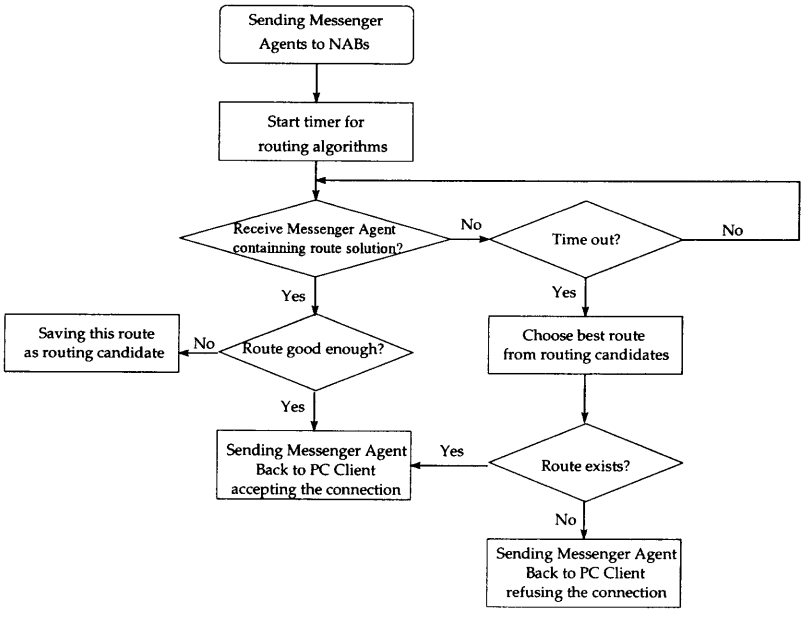


Figure 2.4 Algorithm for Broker Agent in SP to choose a route for its client

As seen by the flow chart in figure 2.4 in order to control the connection setup time a timer is used to determine the deadline of the route searching procedure. If the Broker Agent receives a Messenger Agent from NAB with satisfactory routing solution before the expiration of the timer the route searching process stops and this solution is selected. Otherwise, when the timer expires the Agent chooses the best route among the route candidates found until that time.

Each NAB also creates a Broker Agent to deal with the connection when it receives the Messenger Agents with the corresponding connection request from the SP. Then three kinds of agents are used to implement the routing algorithm as follows:

- *Browser* Agent:

  Browser Agent will be created and sent to nodes inside the individual private network that the NAB belongs to. These agents will collect resource information such as available bandwidth, delay of the link, price of the link etc. In a similar way the Broker Agent in each NAB will also send out Browser Agents to INABs to see if it can take advantage of network resources from other network providers.

- *Daemon* Agent:

  After collecting the necessary resource information, a Daemon Agent containing the GA code (or other preferred algorithm) and resource related information will be created to implement the routing algorithm described in detail in the previous section. Instead of executing the algorithm in each NAB, the Broker Agent sends Daemon Agents to the most suitable nodes inside its private network (e.g. nodes with enough computation resources such as CPU, memory, etc). In this way, we can balance the computation load among nodes in the private networks, if needed.

- *Messenger* Agent:

After the route computation, Daemon Agents will send the results back to the Broker Agent by using a Messenger Agent. This Agent will be forwarded to the Broker Agent in the SP.

As we can see from the above discussion, by using mobile agents we can not only collect the necessary resource information in an efficient way, but we can distribute and balance the computation load inside the networks, and even more importantly we maintain a high level of fault tolerance in the system by running several route searching procedures in different private networks and therefore performing the overall operation in a distributed way. Also, as mentioned above, this routine computing paradigm is very flexible, so that any customer specific algorithm can be embedded in the system without having to upgrade the whole system.

Even though the complete platform and scenario has not been fully deployed yet, we believe that these preliminary measures demonstrate that the overhead introduced to the system by using this agent-based approach is very low. First it should be noted that all the agents used (e.g. browser, daemon, messenger) are very small in size, ranging from 3Kbs to 8Kbs. Therefore the overhead introduced in the system by the agents' size is very low especially when considering the level of flexibility that is obtained by using such an approach. The frequency with which the agents collect data on the network and the specific routes are computed and evaluated, depends on the network itself. Thus it is not possible to set a single value for such parameters, but the administrator may decide the right values after a fine tuning of the relevant network parameters. Regarding the overhead due to the migration of agents some measurements on a 10Mbit LAN were performed, in order to evaluate the order of magnitude of these values [33]. The observed migration time of a 5Kb agent between two points (Pentium III, 350MHz with 128 MB Ram) was 106ms in conditions of light network traffic, which represents a good indication of the concrete applicability of the proposed approach. Taking into account the network configuration depicted in Figure 8, the agents have to migrate from the SP to the three

network providers. This can be done by sending in parallel three agents, minimizing the total migration time; in a more realistic network scenario we do not expect this to be a bottleneck, since as explained earlier, agents do not interfere with the internal routing mechanism of the network provider. They are used only to collect aggregated data of a network provider, which in turn will be used by the genetic algorithm. In general, we believe that the involved migration times are very low with respect to the frequency with which routes are evaluated in the networking environment under consideration.

2.4 Results and Evaluation

In this section we present some numerical results to assess the operation, performance, and efficiency of our algorithm by applying it in some test case scenarios.

2.4.1 Simulation Scenario

Before proceeding with the presentation of the actual numerical results, we briefly describe the networking environment and the corresponding system parameters and service requirements for the test cases we studied. Specifically we assume a multi-operator network with one SP, one CP and 10 nodes belonging to different Network Providers. As mentioned before those 10 nodes represent either the access nodes from SP to the individual networks, or the edge nodes that provide interconnectivity among the networks of the various operators, or the egress nodes that provide connectivity to the CPs. The corresponding topology of the network under consideration in this numerical study is depicted in figure 2.5.

As explained in previous sections a link interconnecting nodes belonging to the same network provider may represent a combination of links interconnecting other intermediate nodes belonging to the same network. In this figure, for sake of simplicity, we only present the nodes of interest (i.e access, egress or edge nodes) and their interconnections. In this figure, for sake of simplicity, we only present the nodes of interest (i.e access, egress or

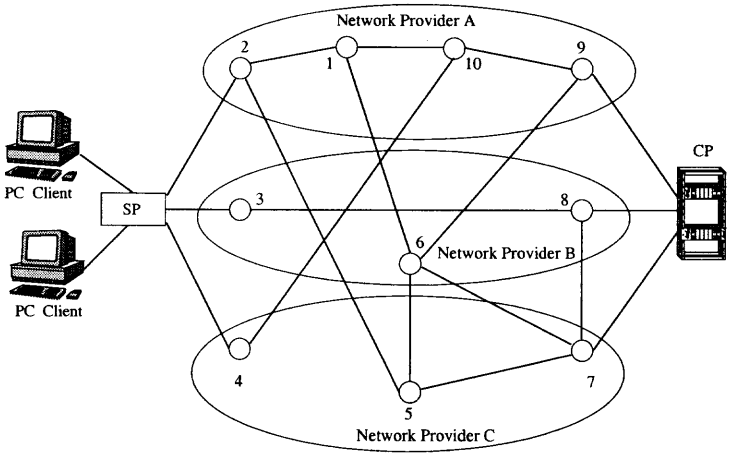


Figure 2.5 Experimental network topology

edge nodes) and their interconnections. In the following we assume that cost and delay of links between any two nodes at certain time period are given. For nodes that no link exists between them or the bandwidth of the link can not support the traffic request between SP and CP, a large delay value(999) is assigned to that link. The cost matrix and delay matrix are given in table2.1 and table2.2 respectively.

99	15	99	99	99	23	99	99	99	99	99
15	18	99	99	23	99	99	99	99	99	99
99	99	15	99	99	99	99	300	99	99	99
99	99	99	14	99	99	99	99	99	99	99
99	23	99	99	99	50	30	99	99	99	99
23	99	99	99	50	99	20	99	18	99	99
99	99	99	99	30	20	99	10	99	30	30
99	99	300	99	99	99	10	99	99	99	15
99	99	99	99	99	18	99	99	99	25	05
99	99	99	99	99	99	99	99	25	99	99

Table 2.1 Cost matrix

In these tables, element (i, j), $i \neq j$, is the cost(delay) from node i to node j, i,j $\in [1, 2, 3...10]$, element (i, i) is the cost(delay) from SP to node i, element $(i, 11)$ is the cost(delay) from node i to CP. The upper-bound(requirement) of time delay (Maxdelay) is assumed to be 25 time units in this study.

999	07	999	999	999	10	999	999	999	05	999
07	03	999	999	06	999	999	999	999	999	999
999	999	01	999	999	999	999	05	999	999	999
999	999	999	02	999	999	999	999	999	10	999
999	06	999	999	999	20	05	999	999	999	999
10	999	999	999	20	999	20	999	07	999	999
999	999	999	999	05	20	999	07	999	09	05
999	999	05	999	999	999	07	999	999	999	01
999	999	999	999	999	07	999	999	999	03	02
05	999	999	999	999	999	999	999	03	999	999

Table 2.2 Delay matrix

The GA related parameters used in our simulation are as follows: population size: 71; crossover rate: 0.6; mutation rate: 0.1; best solution reservation: 5% of population size; MinTrails: 100; MinChangeRate : 20.

2.4.2 Experimental Results

Based on the simulation scenario described in the previous subsection the optimal solution (route) is designed as: [2 5 7 8 0 0 0 0 0 0] and the corresponding cost of this route is 96. According to the selected encoding approach this route actually represents the following path: SP–2–5–7–8–CP. Initially we applied our algorithm without considering the stopping conditions (that is, we let the algorithm run for many iterations) to the network described above. In order to study in depth the behavior of our approach we repeated the experiment many times. The corresponding results are demonstrated in the following figures 2.7–2.11. For most of the runs the algorithm obtains the optimal solution in about 130 steps as demonstrated in figure 2.6. In some cases our algorithm finds the optimal solution very quickly(in about 40 steps) as in figure 2.7, however there may be cases where longer convergence times are observed (figure 2.8) in order to obtain the optimal solution. Furthermore in figure 2.9 we notice that the algorithm has not obtained the optimal solution even after 250 iterations. However even in this case we can see that the cost of the sub-optimal solution obtained by our algorithm (i.e. [4 10 9 0 0 0 0 0 0 0])

is very close to the optimal one, and furthermore as a trade off, our algorithm achieves a solution with lower delay than the optimal one. In any case, as it can be seen by all figures 2.7–2.9 the algorithm always finds a good solution in less than 130 steps favoring the big changes at the beginning if needed, while applying smoother changes to the solution at every step when the intermediate solution obtained is in the neighborhood ("close to") of the optimal solution.

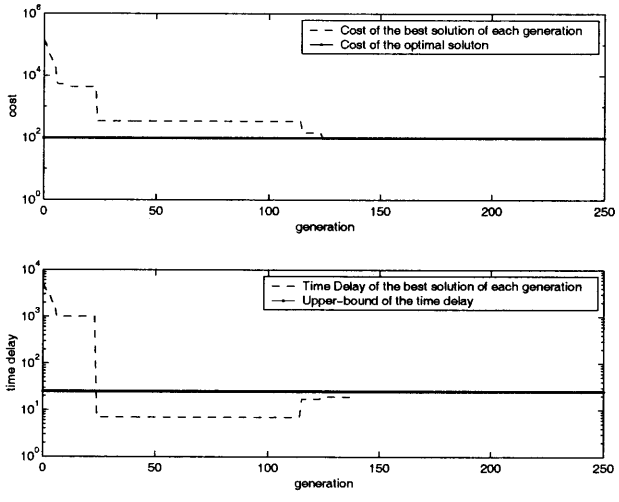


Figure 2.6 GA converges to optimal solution at "middle" speed

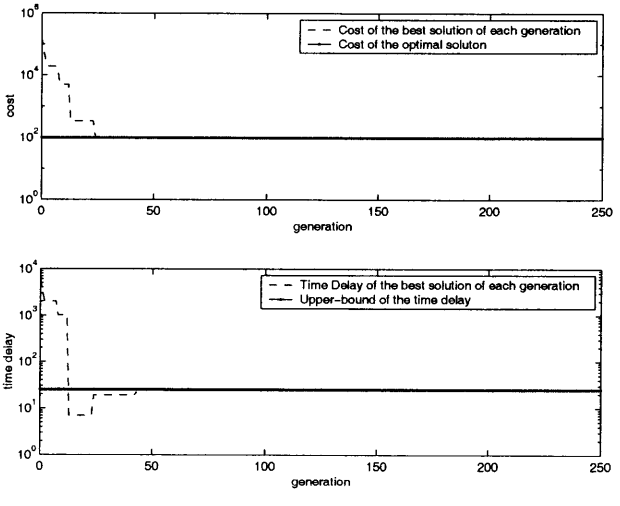


Figure 2.7 GA converges to optimal solution at "high" speed

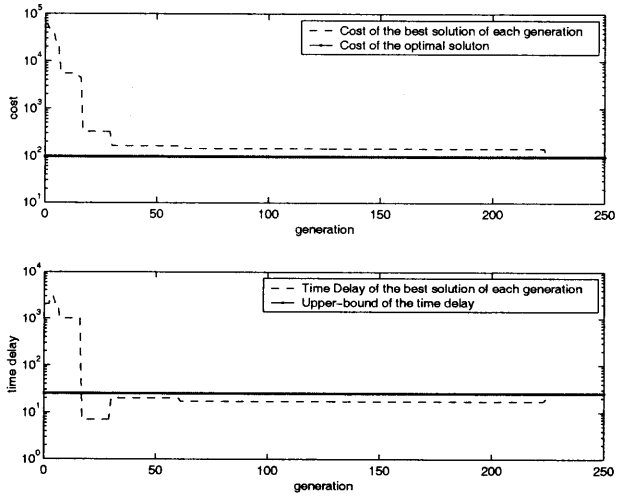


Figure 2.8 GA converges to optimal solution at "slow" speed

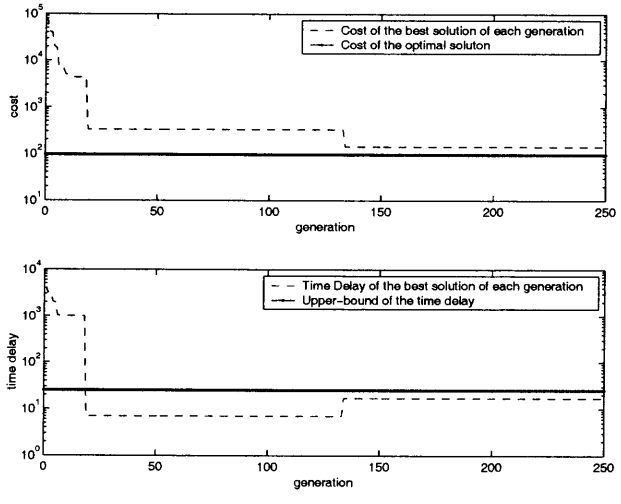


Figure 2.9 GA gets only suboptimal solution

In addition it should be noted that in most of the practical cases the involved parties are mainly interested in obtaining a cost-efficient solution in real-time and not necessarily (especially in cases where the improvement of the optimal solution is very small) the optimal one. Taking this observation into consideration in figures 2.10 and 2.11 we present some numerical results regarding the operation of our algorithm, when we apply the combination stopping conditions described in the previous section. From those results we confirm that after applying the stopping conditions our algorithm stops having identified a cost efficient solution (either optimal or sub-optimal with cost close to the optimal one and lower delay). In fact, in the mobile agent environment, by running several route searching procedures based on GA in different network elements, we considerably improve the probability of getting the optimal routing solution.

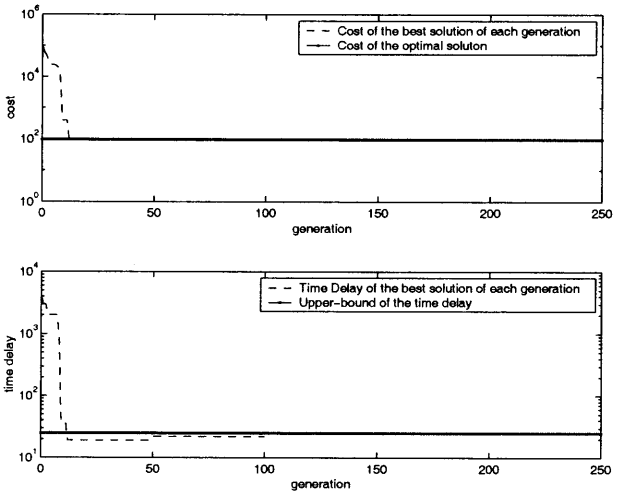


Figure 2.10 GA is stopped after getting optimal solution

2.5 Conclusions

In this chapter we proposed the integration of mobile agents and genetic algorithms in order to provide efficient resource allocation in a multi-operator networking environment. The agent based approach we proposed is used as an effective method for performing basic management operations of a network, such as collection of information about the state

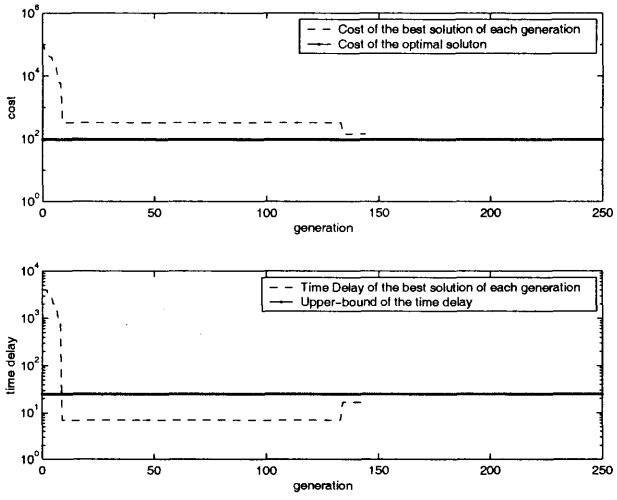


Figure 2.11 GA is stopped after getting suboptimal solution

of the network and the management of network devices through the monitoring of health functions. In the multi-operator network model under consideration in this work, the agent based network management approach represents an underlying framework and structure. To demonstrate the capabilities of such an approach in this environment, we proposed and developed a stochastic algorithm searching process by emulating biological selection and reproduction, in order to identify optimal based routes based on the information provided by the mobile-agent based structure. Using as basis the underlying structure provided by the use of mobile agents, we can not only collect the necessary resource information in an efficient way, but we can distribute and balance the computation load inside the networks, and even more importantly we can maintain a high level of fault tolerance in the system by running several route searching procedures in a distributed way in different private networks. The numerical results reported in this chapter demonstrated that our strategy always identifies in a small number of steps a cost efficient solution that is either the optimal one, or in some cases sub-optimal with cost very close to the optimal and lower delay.

CHAPTER 3

MODELING AND ANALYSIS OF MA BASED NETWORK MANAGEMENT SYSTEM

3.1 Introduction

Since mobile agent technology emerged as a promising paradigm and is much more flexible and dynamic than the Client-Server paradigm, it has recently become an active research and development topic. As mentioned before in most of the research works reported in the literature, the emphasis is placed on the platform used to support the specific management function and qualitative mainly arguments are made regarding the benefits of the use of the agent technology, while very little work has been reported in the area of modeling and analysis of the mobile agent based network management paradigm.

In this chapter, a general analytical model and framework for the evaluation of various network management paradigms is introduced and discussed. It is also illustrated how the developed analytical framework can be used to quantitatively evaluate the performances and tradeoffs in the various computing paradigms, by comparing the performances of the mobile agent based paradigm with the corresponding ones under the Client-Server mode under different scenarios. Specifically in section 3.2 the various entities involved in the network management activities and the possible interactions among them are identified. A network performance monitoring application is used as an example for the later discussion. Based on the interaction model, in section 3.3 various important performance metrics are introduced and obtained analytically. More specifically in subsection 3.3.1 we first describe some performance parameters that are important from the system design point of view, and introduce some notations that will be used throughout our analysis. In subsection 3.3.2 we discuss the analytical evaluation and calculation of the management performance of the CS paradigm, while the corresponding performances in the mobile agent paradigm are discussed in subsection 3.3.3. In section 3.4, we

illustrate how the developed analytical framework can be used to quantitatively evaluate the performances and tradeoffs in the various computing paradigms, by comparing the performances of the mobile agent based paradigm with the corresponding ones under the Client-Server mode under different scenarios. Finally section 3.5 concludes this chapter.

3.2 Management Entities and Interaction Model

In general, the entities involved in network management activities can be classified into three groups, as follows:

- Task Executer (TE). This is the entity that knows how to execute the task and executes the task according to its knowledge in the management activity. For example, the Manager in the Simple Network Management Protocol which generally is located at the management station is the TE.

- Data Provider (DP). This is the entity that owns and provides raw data collected from the network elements. An example is the Agent in SNMP protocol that handles the Management Information Base (MIB) and provides data to the Manager.

- Result Receiver (RR). This is the entity that needs the result generated by the TE. For instance, the management station that gets the network performance reports is the Result Receiver; the network element that updates the system configuration according to the performance monitoring parameters is also the Result Receiver.

Generally the management task is initiated by the TE that plays a core role in the management activity. The TE should be associated with a network device that can provide sufficient computational resources for the management computation purposes. During the management process the TE usually needs some data stored at the individual network devices in the network. Therefore the TE may request the DP at each network device to send the required data. According to different management tasks, this data accessing interaction may repeat many times during the management process. Furthermore,

during the management process, the TE may generate (partial) management results, and these results may be required to be reported to some central management entity, or to be used to manipulate (configure) the network devices managed in the network. If the interaction happens on the same site (i.e. on the same network element), the interaction is a local interaction. If the interaction happens between two entities associated with two different network elements, the interaction is a remote interaction over the network. These interactions in the management task will directly affect the management performance in different management paradigms. For example, the remote interactions between a TE and a DP for data accessing may generate a lot of traffic in the network, while at the same time may introduce significant delay in the management reaction time.

In the following, for the sake of the discussion, a network performance monitoring task and its interaction sequence model is considered as an example. Let us assume that a management station should execute a network performance monitoring task in a certain management domain which includes Q network elements. In this management activity, there are $Q + 1$ Result Receivers: the management station that needs the report of the network performances and Q network elements (node V_1 to V_Q) that need to be re-configured according to the calculation results of TE. Let us also assume that all the data needed for the task are organized in MIBs at Q Management Information Bases $MIB_1, MIB_2, ..., MIB_Q$ located at nodes $V_1, V_2, ..., V_Q$ respectively (these are the Data Providers). In order to carry out the task, the TE, located at node V_0, should access some or all the MIBs to obtain the necessary data to calculate the performances, and/or make some configuration on these nodes. The sequence of the interactions in the task can be described as follows:

```
(0) i = 1;
(1) While (i< Q + 1) {
(2)     TE accesses the $MIB_i$ to get data;
```

```
(3)     TE applies algorithm on the data to calculate the perfor-
        mances;
(4)     if TE need more data from $MIB_i$ (with certain probabili-
        ty)
(5)         go to (2);
(6)     else
(7)         do configuration work at node $V_i$ with probability
            $p_{cfg}$;
(8)         Partial result is reported to Management Station at
            node $V_0$ with probability $p_r$;
(9)         if TE need data from MIB at next node (with certain
            probability)
(10)            i++;
(11)            go to (2);
(12)        else
(13)            finish the task;
(14) }end while
```

At the beginning of the task, the TE needs to access the data stored in the MIB at node V_1. The data access interaction could happen more than once at the same node and the number of data accessing depends on the specific algorithm. Let us assume that the average duration of each MIB access is denoted by t_{MIB} (seconds), the average size of the data accessed each time from the MIB is S_d (bits), while the number of accesses at the same node is distributed with some probability density function (pdf) $f_{sv}(n)$ ($n \in [1, \infty)$). For each batch of data received from the MIB, the algorithm takes on the average t_{alg} seconds to perform the corresponding computation (step (3)). Before completing the task at node V_1, some partial results may be required to be sent back to the Management Station if the the computation is executed remotely. Let us also denote by S_r (bits) the average size of the report and by p_r the corresponding reporting probability. Furthermore let us assume that in general with probability p_{cfg} some configuration work is performed at node V_1 according to the computation result.

After the interactions between the TE and V_1 are completed, the task may need to access the information at some other network element (e.g. V_2) and repeat the data access procedure described above. In the following we assume that the total number of nodes that are involved in the interactions with the TE at node V_0 follows some distribution with pdf

denoted by $f_{dv}(n)$ ($n \in [1, Q]$). All the interactions (data accessing, algorithms applying, configuration etc.) could be local interactions or remote interactions depending on whether or not the task management entity (TE) is at the same node where the data is stored.

3.3 Interaction Mapping and Performance Calculations

3.3.1 Figures of Merit of the Management System and Notations

Before proceeding with the presentation of the performance computation and evaluation of the different computing paradigms we first introduce and describe some performance parameters and corresponding notations that are used throughout this study. In principle the performance metrics that systems designers are interested in can be categorized as follows:

- Network traffic related performances, such as traffic generated around a network devices, total traffic generated in the management activity etc. These traffic related parameters illustrate the overhead introduced by the management application and paradigm to the network, and can be also used to determine the potential bandwidth bottlenecks of the network devices.

- Computational resource usage at certain network elements, such as memory usage and number of instructions executed by a network devices. These parameters are used to determine the computational load of each network device and to measure the scalability of the system.

- Time related performance parameters, such as time used by the management procedure and total time used by the remote interactions between entities on different nodes in the network. These parameters illustrate the reaction speed of the management application and the reliability of the system.

In the following we provide a list with the definition of all the performance parameters and the corresponding notations that are used throughout this chapter:

- S_{total} in bits: total traffic generated during the management task.

- S_{V0} in bits: traffic from/to the node V_0 where the task starter is located.
- S_{Vi} in bits: traffic from/to the node V_i. The traffic to or from a certain node may be used to measure the scalability of the system. The more evenly the traffic load is balanced among the nodes involved in the computation, the higher the scalability of the system is.
- T_{total}: total task execution time used for the management task. It can be used to evaluate the reaction speed of the system.
- T_{rmt}: total remote interaction time. The remote interaction time between any two nodes of the computational task can be used to measure the reliability of the task, and is quite critical especially in a "severe" networking environment such as in satellite networks, ad hoc wireless networks etc.
- M_i in (bits * seconds): memory usage during the management task at the node i.
- I_i: instructions executed at node V_i during the task.
- bw_l^{ij}: bandwidth (bps) of the link between nodes V_i and V_j; (here we assume $bw_l^{ij} = bw_l^{ji}$).
- t_d^{ij}: link delay between nodes V_i and V_j
- S_{MA}: average size (bits) of mobile agent
- S_{TM}: average size (bits) of task manager
- S_q: average size (bits) of a query initiated by the task manager in the Client-Server paradigm
- S_d: average size (bits) of data accessed each time by the task manager
- S_r: average size (bits) of intermediate partial result generated from the interaction between the task manager and node V_i.

- S_{cfg}: average size (bits) of the command and corresponding data used to configure a node.
- t_{MIB}: average time (seconds) for the MIB access.
- t_{alg}: average time (seconds) for processing one batch of data accessed from the MIB.
- N_{sv}: average number of interactions between TE and certain MIB at a node; this value can be calculated from the pdf $f_{sv}(n)$.
- N_{dv}: average number of MIBs at different nodes involved into the interactions with TE; this value can be calculated from the pdf $f_{dv}(n)$.
- p^i_{acs}: probability that the node Vi could be involved into the management task (i.e. its MIB is accessed by the task manager).
- p^i_{cfg}: probability of configuring node V_i after a computation is completed.
- p_r: reporting probability, i.e. the probability that some intermediate partial results may need to be reported to the management station during the management process.

3.3.2 Performance in Client-Server Paradigm

Having defined the interaction model and the performance parameters of interest, we need to map the interaction model to the specific management paradigm. This means that, for the candidate management paradigm, we should determine: a) which entities will be used as the Task Executer, Data Provider or Result Receiver; b) where are the various entities located inside the network; and c) how will these entities interact with each other to implement the task activity. In Client-Server mode, the task manager associated with the management station is the Task Executer that knows how to execute the task; it is also a Result receiver that needs to obtain the performance of each node. The MIBs at Q different nodes are Data Providers and these Q nodes are also the Result Receivers to be re-configured by the management results. The execution of the performance monitoring

task can be mapped into the Client-Server paradigm as follows: the Task Manager at node V_0 sends out a data query (with average size of S_q bits) to the node V_1. A management agent that receives the query accesses the data from the MIB and the data is sent back to the MS. The Task Manager applies the management function on the data and may initiate further data queries according to the interaction model described above. Finally, when Task Manager finishes the computation of the data from node V_1, it generates the partial result. If any configuration is required as a result of that operation, the configuration command (along with the configuration data) is sent to node V_1. If required, the Task Manager may repeat the same procedure with nodes V_2, V_3, etc.

The average traffic from (to) the TE at node V_0 can be calculated as follows:

$$\begin{aligned} S_{V0}^{CS} &= [(S_q + S_d)N_{sv} + S_{cfg}p_{cfg}]N_{dv} \\ &= [(S_q + S_d)\sum_{n=1}^{\infty} nf_{sv}(n) + S_{cfg}p_{cfg}]\sum_{n=1}^{Q} nf_{dv}(n) \end{aligned} \tag{3.1}$$

Correspondingly, the traffic from (to) node $V_i (i \in [1, Q])$ is:

$$\begin{aligned} S_{Vi}^{CS} &= [(S_q + S_d)N_{sv} + S_{cfg}p_{cfg}]p_{acs}^{i} \\ &= [(S_q + S_d)\sum_{n=1}^{\infty} nf_{sv}(n) + S_{cfg}p_{cfg}]\sum_{n=i}^{Q} f_{dv}(n) \end{aligned} \tag{3.2}$$

The average number of instructions executed at node V_0 is given by:

$$I_0^{CS} = mN_{sv}N_{dv} = m\sum_{n=1}^{\infty} nf_{sv}(n)\sum_{n=1}^{Q} nf_{dv}(n) \tag{3.3}$$

while for any other node V_i, $i = 1, 2...Q$ we have: $I_i^{CS} = 0$.

The average execution time for the task is:

$$
\begin{aligned}
T_{total}^{CS} &= \sum_{i=1}^{Q} T_{total}^{i} p_{acs}^{i} \qquad (3.4)\\
&= \sum_{i=1}^{Q} \{[(\frac{S_q}{bw_l^{0i}} + t_d^{0i} + t_{MIB} + t_{alg} + \frac{S_d}{bw_l^{0i}} + t_d^{0i}) N_{sv} \\
&\quad + (\frac{S_{cfg}}{bw_l^{0i}} + t_d^{0i}) p_{cfg}] \sum_{n=i}^{Q} f_{dv}(n)\} \\
&= \sum_{i=1}^{Q} \{[(\frac{S_q}{bw_l^{0i}} + t_d^{0i} + t_{MIB} + t_{alg} + \frac{S_d}{bw_l^{0i}} + t_d^{0i}) \sum_{n=1}^{\infty} n f_{sv}(n) \\
&\quad + (\frac{S_{cfg}}{bw_l^{0i}} + t_d^{0i}) p_{cfg}] \sum_{n=i}^{Q} f_{dv}(n)\} \\
&= \sum_{i=1}^{Q} \{[(\frac{S_q + S_d}{bw_l^{0i}} + 2t_d^{0i} + t_{MIB} + t_{alg}) \sum_{n=1}^{\infty} n f_{sv}(n) \\
&\quad + (\frac{S_{cfg}}{bw_l^{0i}} + t_d^{0i}) p_{cfg}] \sum_{n=i}^{Q} f_{dv}(n)\}
\end{aligned}
$$

The average remote interaction time is derived by subtracting the local interaction time from the average execution time, as follows:

$$
\begin{aligned}
T_{rmt}^{CS} &= \sum_{i=1}^{Q} \{[(\frac{S_q}{bw_l^{0i}} + t_d^{0i} + \frac{S_d}{bw_l^{0i}} + t_d^{0i}) N_{sv} + (\frac{S_{cfg}}{bw_l^{0i}} + t_d^{0i}) p_{cfg}] \sum_{n=i}^{Q} f_{dv}(n)\} \qquad (3.5)\\
&= \sum_{i=1}^{Q} \{[(\frac{S_q + S_d}{bw_l^{0i}} + 2t_d^{0i}) \sum_{n=1}^{\infty} n f_{sv}(n) + (\frac{S_{cfg}}{bw_l^{0i}} + t_d^{0i}) p_{cfg}] \sum_{n=i}^{Q} f_{dv}(n)\}
\end{aligned}
$$

The average memory used at node V_0 for computation is:

$$
M_0^{CS} = (S_{TM} + QS_r) T_{task} \qquad (3.6)
$$

where, S_{TM} is the size (bits) of the task manager, while the corresponding average memory used at the other nodes V_i, $i = 1, 2...Q$, is obtained as follows:

$$
\begin{aligned}
M_i^{CS} &= S_d (T_{MIB} + \frac{S_d}{bw_l^{0i}}) N_{sv} p_{acs}^{i} \qquad (3.7)\\
&= S_d (T_{MIB} + \frac{S_d}{bw_l^{0i}}) \sum_{n=1}^{\infty} n f_{sv}(n) \sum_{n=i}^{Q} n f_{dv}(n)
\end{aligned}
$$

Therefore the total cost of the task (including the memory usage and the communication traffic cost) is:

$$C_{total}^{CS} = w_1 S_{total}^{CS} + w_2 M_{total}^{CS} = w_1 \sum_{i=0}^{Q} S_{Vi}^{CS} + w_2 \sum_{i=0}^{Q} M_i^{CS} \quad (3.8)$$

where w_1 and w_2 ($w_1 + w_2 = 1$) are the weights for the communication cost and for the memory usage respectively.

3.3.3 Performance in MA Paradigm

In mobile agent mode, a Mobile Agent containing the management algorithm is used as the Task Executer. Instead of moving the data to the node V_0 to perform the computation, the MA is sent out (with size S_{MA}) that contains the computation algorithm to nodes V_1,... ,V_Q in order to access the necessary data and apply the algorithm on the data locally. Then, the interaction between the TE and the DP becomes local interaction. On the other hand, the result generated at the node be managed should be sent back to the management station cross the network so the result reporting operation becomes a remote interaction in this case. However, if configuration is required at the node, the configuration becomes a local interaction. If the computation needs further data from other nodes, the MA will move to the next node where the data is located. Along with the MA, the partial result may also be required at the next node. If at some node V_i, the task is completed, then a final result is generated and the MA may be "killed".

Under this scenario the average traffic from (to) node V_0 where the task "starter" is located is:

$$S_{V0}^{MA} = S_{MA} + (S_r p_r \sum_{n=1}^{Q} n f_{dv}(n)) \quad (3.9)$$

This traffic includes sending out the entity of the MA and receiving the results (or partial results) from all the nodes involved in the task.

The average traffic from (to) node $V_i (i \in [1, Q-1])$ is

$$S_{Vi}^{MA} = (S_{MA} + S_r p_r) \sum_{n=i}^{Q} f_{dv}(n) + (S_{MA} + S_r) \sum_{n=i+1}^{Q} f_{dv}(n) \tag{3.10}$$

The traffic here includes getting the MA from some other node (or the starter) and sending the MA to the next node. It also includes the traffic due to the result reporting mechanism (with probability p_r) as well as due to the fact that the partial result may be used in next node.

For node V_Q (last node in the execution sequence), since the task completes at that node and the final result will be reported back to node V_0, the corresponding traffic is:

$$S_{VQ}^{MA} = (S_{MA} + S_r) \sum_{n=i}^{Q} f_{dv}(n) \tag{3.11}$$

It should be noted here that there are no operations (algorithms) required to be executed at node V_0 and therefore $I_0^{MA} = 0$, while the average instructions executed at other nodes $V_i (i \neq 0)$ are given by:

$$I_i^{MA} = m N_{sv} p_{acs}^i = m \sum_{n=1}^{\infty} n f_{sv}(n) \sum_{n=i}^{Q} f_{dv}(n) \tag{3.12}$$

The average execution time can be calculated as:

$$T_{total}^{MA} = \sum_{i=1}^{Q} T_{total}^i p_{acs}^i \tag{3.13}$$

where, T_{total}^i includes the time for the MA (along with partial results at previous node) transportation from the last node, the computation time at node V_i, and the time for the result reporting operation. The time T_{total}^i for node V_i is obtained as follows:

$$T^i_{total} = \frac{S_{MA}+S_r}{bw_l^{i-1,i}} + t_d^{i-1,i} + (t_{MIB}+t_{alg})\sum_{n=1}^{\infty} nf_{sv}(n) + (\frac{S_r}{bw_l^{0i}} + t_d^{0i})p_r \qquad (3.14)$$

Therefore we can easily conclude that:

$$T^{MA}_{total} = \sum_{i=1}^{Q}\{[\frac{S_{MA}+S_r}{bw_l^{i-1,i}} + t_d^{i-1,i} + (t_{MIB}+t_{alg})\sum_{n=1}^{\infty} nf_{sv}(n) + (\frac{S_r}{bw_l^{0i}} + t_d^{0i})p_r]\sum_{n=i}^{Q} f_{dv}(n)\} \qquad (3.15)$$

The average remote interaction time can be derived by subtracting the local interaction time from the average total task time T^{MA}_{total}, as follows:

$$T^{MA}_{rmt} = \sum_{i=1}^{Q}\{[\frac{S_{MA}+S_r}{bw_l^{i-1,i}} + t_d^{i-1,i} + (\frac{S_r}{bw_l^{0i}} + t_d^{0i})p_r]\sum_{n=i}^{Q} f_{dv}(n)\} \qquad (3.16)$$

The average memory used at node V_0 is:

$$M_0^{MA} = S_{MA}(\frac{S_{MA}}{bw_l^{0,1}} + t_d^{0,1}) + QS_rT_{task} \qquad (3.17)$$

Regarding the memory used at node $V_i, i \neq 0$, the mobile agent only stays at a node as long as the interaction is needed. After that the MA will automatically move to next node if the task is not yet completed, or kill itself if the task is completed. Thus the memory used at node V_i to store the MA itself, the data from the MIB and the partial result during the interaction is given by the following expression:

$$M_i^{MA} = (S_{MA}+S_d+S_r)(T^i_{total}p^i_{acs} + \frac{S_{MA}+S_r}{bw_l^{i,i+1}}p^{i+1}_{acs}) \qquad (3.18)$$

Based on the above results the total cost of the task can be obtained as:

$$C^{MA}_{total} = w_1S^{MA}_{total} + w_2M^{MA}_{total} = w_1\sum_{i=0}^{Q} S^{MA}_{Vi} + w_2\sum_{i=0}^{Q} M_i^{MA} \qquad (3.19)$$

3.4 Discussions and Numerical Results

In this section we discuss how the models and results obtained in the previous section can be used to gain some insight about the use of different management architectures and computing paradigms under different networking management tasks. We first compare the performances of the client server paradigm and the mobile agent based architecture, and discuss their applicability as well as their tradeoffs. We present some numerical results for different networking scenarios that demonstrate the applicability of our proposed framework, quantify the corresponding tradeoffs involved, and provide guidelines about the conditions that the MA based approach outperforms the traditional CS approach.

3.4.1 Client-Server Paradigm v.s. Mobile Agent Paradigm

In this section we compare the corresponding performances of the client server and the mobile agent paradigms from two different aspects that are very critical for the effective and efficient operation of the network management architectures, that is the scalability and reliability. Based on the expressions obtained in the previous section we conclude that the scalability of the Client-Server paradigm is mainly affected by the traffic and computational load generated around the management station (node V_0). From relation (3.1) we can see that the traffic around the management station is mainly generated by the data accessing from the MIBs of the nodes in the task, and it is proportional to the average number of nodes from which the data should be fetched. Also the traffic around the management station increases with the increase of the number of data accessing at the same network element. Moreover, we observe from (3.3) that since nearly all the computations are executed at the network management station the computational load is proportional to the number of nodes that are managed as well as the number of data accessing at a node. On the other hand, in the MA paradigm, the traffic load around the management station is mainly generated by the reporting of the results from the various network elements. Therefore, the difference (decrease) of the traffic load around the management station under the MA paradigm and

the CS paradigm can be calculated as follows:

$$\begin{aligned} & S_{V0}^{CS} - S_{V0}^{MA} \qquad (3.20) \\ = \; & [(S_q + S_d)\sum_{n=1}^{\infty} nf_{sv}(n) + S_{cfg}p_{cfg}]\sum_{n=1}^{Q} nf_{dv}(n) - S_{MA} - (S_r p_r \sum_{n=1}^{Q} nf_{dv}(n)) \\ = \; & [(S_q + S_d)\sum_{n=1}^{\infty} nf_{sv}(n) + S_{cfg}p_{cfg} - S_r p_r]\sum_{n=1}^{Q} nf_{dv}(n) - S_{MA} \end{aligned}$$

Under the assumption that the size of the MA is negligible compared to the total network management traffic, the traffic around the network management station is reduced significantly, since in general the raw data transportation from the NEs is much higher than the traffic generated by the reporting of the results from the NEs. At the same time, in the MA paradigm the computational load at the MS is reduced nearly to zero because the management task is executed *locally* on the NEs involved in the task. Therefore under the MA paradigm the computational load locally at the NEs may increase, however the MS is no longer a bottleneck from the point of view of both traffic and computational load, and thus overall it presents a more scalable architecture.

The total management time of a management task can be used to measure the management reaction of the management system. Comparing the total management time of a task under the two different paradigms (relations (3.4) and (3.15)), and assuming that the configuration command in the Client-Server paradigm has the same size and operation probability as that of the partial result, the corresponding difference can be calculated as follows:

$$\begin{aligned} & T_{total}^{CS} - T_{total}^{MA} \qquad (3.21) \\ = \; & \sum_{i=1}^{Q}\{[(\frac{S_q + S_d}{bw_l^{0i}} + 2t_d^{0i})\sum_{n=1}^{\infty} nf_{sv}(n) - \frac{S_{MA} + S_r}{bw_l^{i-1,i}} - t_d^{i-1,i}]\sum_{n=i}^{Q} f_{dv}(n)\} \end{aligned}$$

From (3.21), we can easily see that the difference may increase with the increment of the number of MIB accesses at a certain network. The reason is that in the MA paradigm MIB accessing becomes a local interaction between the management entity and managed entity,

instead of a remote interaction in the CS paradigm. The tradeoff is that the management entity (mobile agent) should be transported among the network elements. Once these parameters are determined for the specific task and scenario under consideration, the difference in the remote interaction time of a management task in the two paradigms can be obtained and evaluated quantitatively.

3.4.2 Numerical Results

In the following we consider a network station that has a number of nodes in its management domain. In order to perform some management task (e.g. consider a network monitoring application used to detect the abnormal status created by some malicious intrusion), we assume that the management station needs to check the MIB in every node in its domain, one after another, according to some certain algorithm (e.g. anomaly detection algorithm). After accessing the data from a MIB, some algorithm is applied on the data to compute certain parameters. The partial results are then used for the computation in the next node. In this scenario, we assume that all the nodes in the domain should be processed by the management station and the final result should be reported to the management station. The values of the related parameters are: data request command $S_q = 50$ bytes; (partial) result $S_r = 200$ bytes; result reporting probability $p_r = 1$; the size of the mobile agent S_{MA} is assumed to be 3Kbytes while the size of the data S_d from MIBs is set as α times the size of the MA ($S_d = \alpha S_{ma}$); the bandwidth available between any pair of nodes is assumed to be 512K bps and the corresponding propagation delay 0.001s.

Figure 3.1 compares the traffic generated around the management station in different paradigms for different data sizes (i.e. different values of α), by using formulas (3.1) and (3.9), as a function of the number of nodes involved in the management task. It can be seen from this figure that the traffic around the management station is fixed in the MA paradigm, no matter what the size of the data from the MIBs is, or how many nodes exist within the domain. The reason is that in the MA paradigm, the traffic around the management station

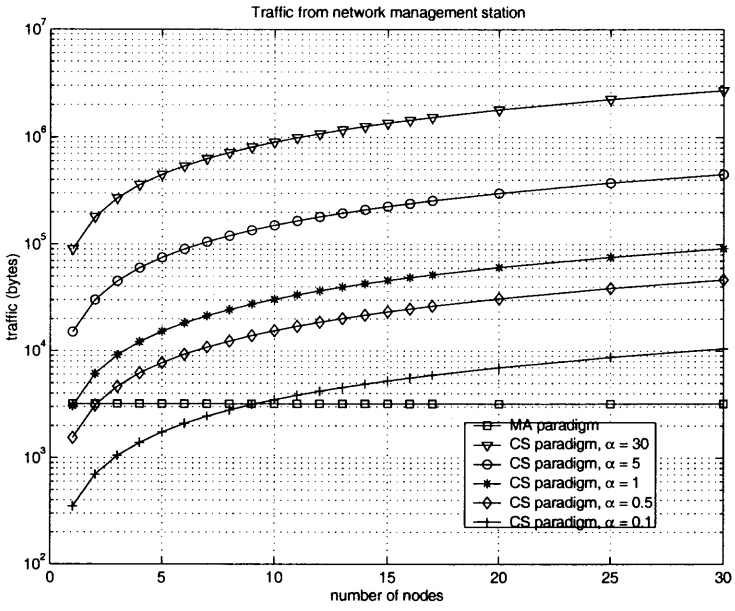


Figure 3.1 Traffic generated around the management station

is due to the initial mobile agent sent from the MS to the first node to be managed plus the final report from the last node involved in the management task to the MS. In general a mobile agent has small size ranging from 3kb to 8kb depending on its functionality ([33]). Therefore in most cases, as it can be seen from figure 3.1 (for values of α larger than 1), the MA approach outperforms significantly the CS mode. However, if the data from the MIBs have smaller size than that of the mobile agent and the system contains a small number of nodes, the Client-Server paradigm could outperform the MA paradigm.

Figure 3.2 compares the total traffic generated in the network by the different management paradigms, and figure 3.3 compares the management reaction time (total management time) for of the management task. As expected the total traffic generated and the management reaction time increase as the number of nodes in the system increase, while regarding the ratio between the data size and mobile agent size the same observations described above are confirmed here as well.

In the following we consider a network with twenty nodes, where we vary the bandwidth between any node and the management station from 0.65 to 1 times 512kbps.

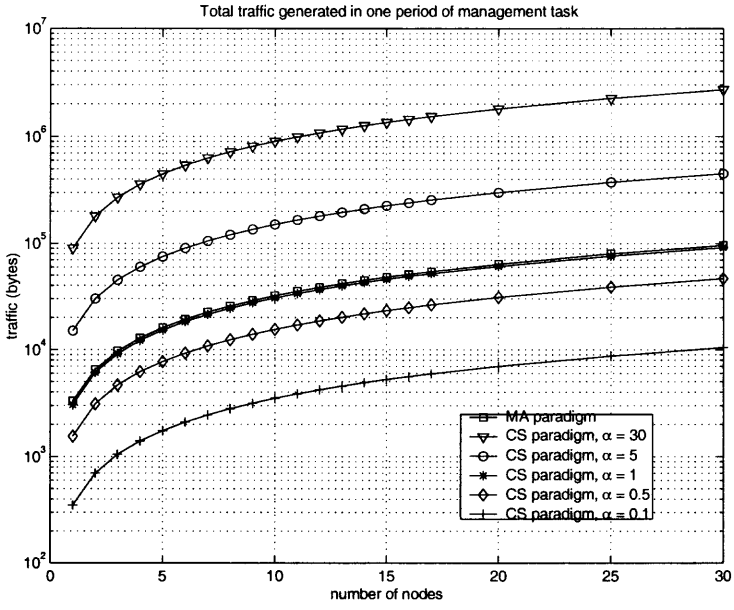


Figure 3.2 Total generated traffic in management task

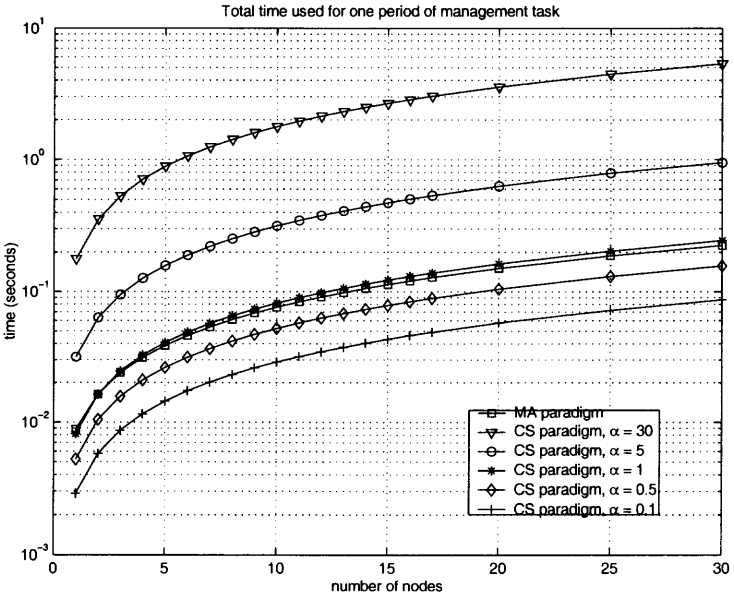


Figure 3.3 Management reaction time

In figure 3.4 we present the management reaction time as a function of the available bandwidth for different management paradigms and different values of α. As shown in figure 3.4, when the size of data S_d is larger than the size of the MA the variation of available bandwidth between the management station and the managed nodes affect the task reaction in the CS paradigm, while the corresponding time is not affected in the MA paradigm (for instance see curves with α equal to 30 in figure 3.4). The reason is that in CS paradigm, a lot of the data from the MIBs should be transported from the managed nodes to the central management station, while in MA paradigm, the MIB accesses are local interactions between the mobile agent and the managed nodes. For the smaller data size scenarios, the impact of the bandwidth variation on the reaction time of the CS paradigm is much smaller.

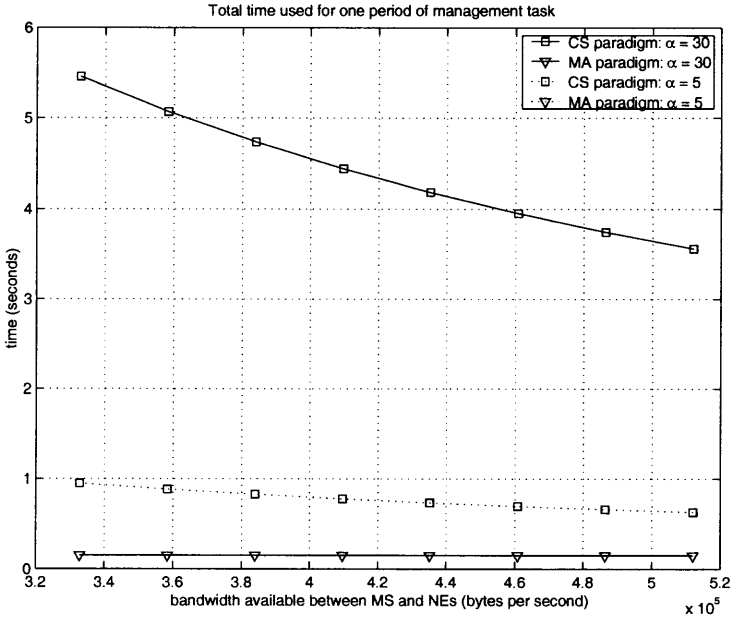


Figure 3.4 Management reaction time vs. bandwidth variation

Next we study the impact of the propagation delay on the achievable performance. In some networking environments, such as the satellite networks, the management station may be located far away from the managed nodes, and therefore the propagation delay between the management station and the remote nodes could vary significantly. In figure 3.5 we

present the result of the management reaction time for both paradigms for different values of ratio α, as a function of the propagation delay. Again we observe that the performance in the MA paradigm is not affected by the variation of the propagation delay between the managed node and the management station. The reason is that the link that experiences high propagation delay will only be used for the initial mobile agent dissemination and the final result reporting.

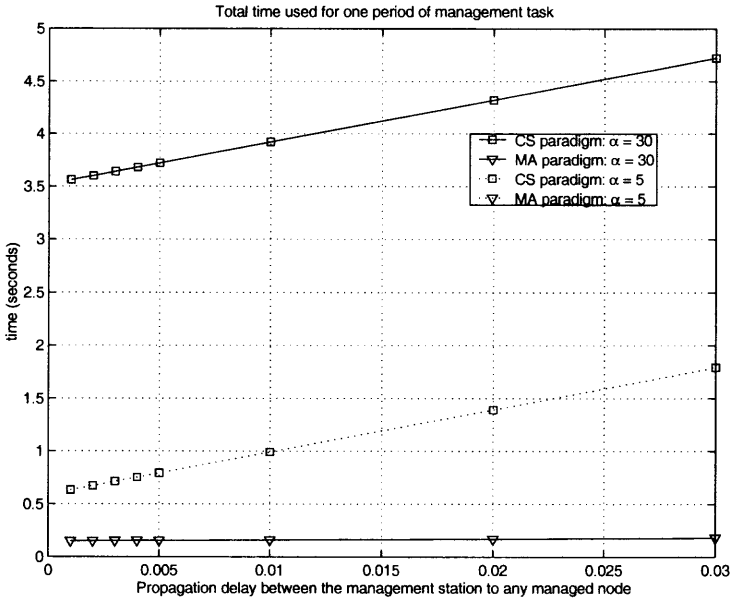


Figure 3.5 Management reaction time vs. propagation delay

3.5 Conclusions

Mobile Agent technology has been recently used as the basis for the design and development of reliable, scalable and flexible architectures for the management of large scale distributed systems. In this chapter a generic framework that can be used for the evaluation and analysis of the performance and tradeoffs of the MA management paradigm is presented and analyzed. Although the emphasis of this chapter is placed on the calculation of the achievable performances under the mobile agent based management strategy, for comparisons mainly purposes the corresponding performances under the

CS mode are also studied and obtained for different scenarios. The developed models and framework are used to gain some insight about the use of different management architectures and computing paradigms under different networking management tasks. Specifically we compare the performances of the client server paradigm and the mobile agent based architecture, and present some numerical results for different networking scenarios that demonstrate the applicability of our proposed framework, quantify the corresponding tradeoffs involved, and provide some guidelines about the conditions that the MA based approach outperforms the traditional CS approach. Hopefully this analytical model could help the system designer in methodology to choose the right management paradigm for a specific management task.

CHAPTER 4

MOBILE AGENT BASED RESOURCE MANAGEMENT IN WIRELESS NETWORK

4.1 Introduction

In wireless networks, one of the most important network management processes is the bandwidth resource management, which aims to increase the bandwidth utilization while maintaining the QoS requirements of users. Since the most significant feature of the mobile cellular network is that the users are moving, the resource management process for wireless networks is much more complicated than that of wireline networks. In fact, when a mobile user moves from one location to another the data path between the source and destination changes. If the cell which the mobile host moves into is overcrowded the available resources may not be sufficient to guarantee the QoS required by the user and as a consequence service might be interrupted.

For some cases where service with strict QoS requirements that are not affected by user movements is required, in order to guarantee the quality of such services, resource reservation needs to be performed at each cell that the mobile user is moving to, as the system evolves. On the other hand for some other cases where the mobile user could accept service with flexible QoS requirements (e.g. the user could specify a set of acceptable QoS levels ranging from a minimum to a maximum), the system may try to rearrange the resource allocation in the new cell in such a way to satisfy the QoS requirements of all the mobile users in the cell.

Although the out-coming new radio technology will bring more bandwidth at air interface, if not managed properly, the bandwidth resources will not meet the requirements of future users. These trends, along with the convergence of communications, information, commerce and computing, are creating a significant demand and opportunity for multimedia personal communication services. Flexibility is a key design issue for the

future wireless network architectures in order to adapt instantly to the changing customer service demands and the dramatic explosion of the number of nodes in a network. Emerging services and new management functionalities may be added in the future and this requires the network to be more programmable to accommodate new features at low upgrade cost. At the same time the growing scale of wireless networks is a big challenge for nowadays network management frameworks. Recognizing this trend and need, in this chapter, we introduce and design a management framework, based on Mobile Agent technology, that supports, with flexibility and efficiency, the functionality of resource management, and thus can meet the objective of providing flexible QoS-enabled seamless multimedia services to mobile users in next generation mobile communication systems. Furthermore based on the model and framework described in chapter 3, we gain some insight about the performance of the above mentioned resource management system, under the Client-Server and the Mobile Agent paradigms.

The main principles of our approach are summarized as follows:

- the system supports multiple classes of service that require various levels of quality that may range from strict to flexible and soft QoS requirements;
- advanced bandwidth reservation is performed to the cell where a mobile user is moving towards in order to assist and support seamless handoff process;
- user mobility is introduced into the reservation scheme and process in order to optimize the efficiency of handoff mechanisms and minimize, if not eliminate, the unnecessary reservation of resources and therefore improve the system capacity and throughput;
- bandwidth reconfiguration processes are developed that may allow the efficient resource re-distribution in a cell to satisfy the QoS requirements of all the mobile users in the cell, especially when users with flexible QoS requirements are supported in the system;

- a mobile agent based framework is proposed and designed to facilitate the efficient implementation of the above integrated approach.

The remaining of this chapter is organized as follows. In section 4.2 the integrated resource management approach for wireless networks is proposed and described. Specifically we first describe the handoff/bandwidth reservation problem and QoS management in wireless networks; then the proposed mobility-assisted bandwidth reservation scheme used in this study is summarized; finally the admission control algorithm and resource allocation procedures used for new calls and handoff calls are described in detail. In section 4.3 the performances of the management system under two different management paradigms are evaluated, while based on the conclusions of this evaluation a mobile agent based system that can be used as the framework for the implementation of the proposed efficient resource reservation mechanisms and call admission control is proposed and designed in section 4.4. The performance analysis of the proposed integrated scheme is provided in section 4.5, while 4.6 concludes this chapter.

4.2 Integrating Bandwidth Reconfiguration with Bandwidth Reservation for Flexible QoS Management

Most of the existing solutions for handling handoff traffic in wireless mobile networks are proactive in nature (e.g. advanced/proactive bandwidth reservation). In general different schemes, that may range from static to dynamic, can be used to implement proactive bandwidth reservations. The underlying principle of these approaches is to reserve some resources for the exclusive use of handoff traffic, while the rest can be shared by all traffic. The key factor that influences the performance of such schemes is how to determine the amount of bandwidth to be reserved. If the reservation is too low then the QoS requirements on handoff traffic can not be met, while if the reservation is too high then resources are wasted and new traffic may be blocked from the system. Similarly if the reservation can

not be used soon, it may result in a waste of the limited radio resources. A brief review of the bandwidth reservation solutions can be found in [34].

Recently some adaptive bandwidth management solutions for multimedia services in wired and wireless networks have been proposed (e.g, [21], [35], [36]). The use of programmable agents for flexible QoS management in wired networks has been demonstrated in [21], where traffic with strict QoS requirements can "borrow" bandwidth from traffic which present less strict QoS requirements. By extending this idea into wireless networks, we develop a call admission/bandwidth reconfiguration scheme which along with the use of advanced bandwidth reservation mechanisms could meet the QoS requirement of real-time traffic and handoff calls in next generation wireless networks.

In general we envision an environment where many classes of service, each one with different QoS requirements and possibly multiple QoS dimensions, could co-exist in a system with a finite set of resources [37]. In the event of the occurrence of heavy load or congestion, the system may not be able to provide the highest QoS to all the various services, since the available resources may not be sufficient. Hence the available resources should be allocated and (re)distributed to the various services according to their respective QoS requirements and priorities. In this study the resource under consideration is the available network bandwidth.

4.2.1 Model and Assumptions

In the following, we assume that there are several classes of traffic in the network and that they have different QoS requirements and priorities [37]. Let Q_{j1}, Q_{j2},...,Q_{jd_j} denote the QoS dimensions of traffic j and P denote the priority space. The QoS dimensions could be any QoS parameters such as average bandwidth, packet loss ratio, delay, jitter etc. Each dimension Q_{ji} is a finite set of quality choices for traffic j's i^{th} QoS dimension, where q_{ji}^k denotes the k-th choice of traffic j's i^{th} QoS dimension. These choices can be ordered in sequences $q_{ji}^0, q_{ji}^1, q_{ji}^2, ..., q_{ji}^m$ from minimum to maximum requirement. Thus

each traffic j with certain level-k QoS and priority p can be characterized using the tuple $((q_{j1}^k, q_{j2}^k,...,q_{jd_j}^k), p)$. The concept of effective bandwidth (e.g.[38, 39, 40, 41]) can be used to represent the amount of bandwidth required by each class of service to statistically achieve the desired QoS. Effective bandwidth is a scalar that summarizes resource usage and that depends on the statistical properties and QoS requirements of a source. Based on this concept, the corresponding bandwidth required by a source-j to achieve its level-k QoS requirement, can be represented as: $Bw_j = F(q_{j1}^k, q_{j2}^k, ..., q_{jd_j}^k)$,where F is a measurement function of resource (bandwidth) usage in order to achieve the desired QoS (e.g. [38]). The actual estimation method of the required effective bandwidth of a class of service is beyond the scope of this dissertation.

By considering the various combinations of different QoS requirements, we can order these combinations according to the value of effective bandwidth. Apparently the combination with all the minimum requirements in every QoS dimension uses the minimum effective bandwidth and the combination with all the maximum requirements in every QoS dimension uses the maximum effective bandwidth. These two values represent the lower bound and upper bound of the required bandwidth of that traffic as follows:

$$Bw_j^{min} = F(Q_j^0) = F(q_{j1}^0, q_{j2}^0, ..., q_{jd_j}^0) \tag{4.1}$$

$$Bw_j^{max} = F(Q_j^{max}) = F(q_{j1}^m, q_{j2}^m, ..., q_{jd_j}^m) \tag{4.2}$$

By ordering the various QoS combinations of traffic j by means of effective bandwidth, these combinations form a totally ordered set $E_j = (Q_j^0, Q_j^1, ..., Q_j^{max})$. For a service j with very strict QoS requirements, set E_j could have only one element with the desired QoS, while in the general case of supporting flexible QoS requirement space, each traffic j with a given priority level, is assumed to accept any QoS level ranging from the minimum QoS requirement Q_j^0 to the maximum QoS requirement Q_j^{max} in its own QoS combination set.

The basic underlying principles of our CAC scheme for the case of multiple classes of service with different priorities are as follows:

- Handoff calls have always higher priority than new calls. Handoff calls may borrow bandwidth from other traffic in case of bandwidth shortage, while new calls do not.
- High priority traffic has higher priority than lower priority traffic to access bandwidth resources in the handoff procedure. In case of bandwidth shortage it can "borrow" bandwidth from lower priority (or with the same priority) traffic currently in the target cell that it handoffs into.
- When lower priority traffic is lending its bandwidth to higher priority traffic, it always attempts to adapt to the immediate lower QoS level, next to the current QoS level, in its QoS combination set $(Q_j^0, Q_j^1, ..., Q_j^{max})$.

4.2.2 Advanced Bandwidth Reservation Mechanism

As mentioned before, our proposed integrated resource management scheme includes two main components: an advanced bandwidth reservation scheme and the bandwidth reconfiguration mechanism. In this subsection, we provide an overview of the first component used in our system, namely the Predictive Mobility-based Bandwidth Reservation Scheme (PMBBR) which can be used to determine the amount of bandwidth to be reserved in each cell. PMBBR is an improved version of Mobility-based Bandwidth Reservation Scheme (MBBR) which has been proposed in [42]. The basic idea of MBBR scheme is that the cell traffic are actually not independent with each other because of the handoffs: when a call enters a cell, it does not only consume bandwidth in the current cell, but also generates requirements on the resources in the neighboring cells by certain probability. In other words, ongoing calls in the current cell will exert some influence on the bandwidth assignment in the neighboring cells. Through the introduction of influence

curves, the MBBR scheme quantifies the influence based on the statistical user mobility information such as moving direction and cell dwell time.

Let us assume that each mobile station has the capability to detect its location in the cell by certain means such as Global Position System. Based on the history location information, various prediction algorithms can be used to predict the future moving speed and direction of each user. In the PMBBR scheme, a 2-order autoregressive process model [43] is used for the prediction purposes. With these predictions, the MBBR scheme can be improved in the following aspects:

- The directional factors. In MBBR scheme, the directional factors are statistical values for all the users in current cell, which means that the influence value of each single user is distributed to all the neighbor cells according to directional factors. With the predicted moving direction and current position, we can accurately calculate which is the handoff target cell, so that the reservation is made only in one cell, and therefore the resource waste in other neighbor cells can be eliminated.

- The handoff probability. Let us denote by t_k^{hf} the time interval in the future that the user k under consideration needs to request a handoff, which can be calculated based on the current position of the user and the predicted moving speed and direction. Then the probability that this ongoing call will request a handoff in the future can be calculated as:

$$\begin{aligned} P_k &= \Pr(\text{this call will request a handoff}) \\ &= \Pr(\text{the residual call lifetime is longer than } t_k^{hf}) = \int_{t_k^{hf}}^{\infty} f(\tau)d\tau \end{aligned} \quad (4.3)$$

where $f(\tau)$ is the residual call life time probability density function. If we assume that call life time is exponentially distributed with mean value T_l, the above probability equals: $e^{-t_k^{hf}/T_l}$

- A critical issue that influences the performance of a bandwidth reservation mechanism is the actual time that the reservation is made for the incoming handoff calls. If the reservation is made at the time that it can be used at the near future, such a scheme can achieve a better performance. Otherwise, the reservation could result in waste of resources and the system will incur unnecessary new call blocking. This problem is also addressed in [44], in which the concept of threshold distance (TD) is introduced to reduce the likelihood of false reservations: users inside the TD circle will not submit reservation requests. But since users may have different moving speed, a high speed user that is currently located inside the TD circle may move out of the coverage region of the cell earlier than a low speed user that is outside the TD circle. Therefore, distance alone is not a good solution to the problem of reservation timing. Based on these considerations, a reservation *advance time* t_{thre} is set. Reservations are made only for those calls which will request a handoff in the near future, *i.e.* $t_k^{hf} \leq t_{thre}$.

Taking all the above factors into consideration, the PMBBR scheme determines the total amount of bandwidth to be reserved in cell j as:

$$R_j = D \sum_{i \in N_j} \sum_{k \in S_i'} BW_k P_k \tag{4.4}$$

where BW_k is the bandwidth requirement of the ongoing call k, and S_i' is the set of those ongoing calls which are currently in cell i and, according to the prediction, are going to handoff to cell j within time t_{thre}. Notice that, because the predictive mobility information is used to calculate the bandwidth reservation, as the number of mobile users in the network changes and the location history of each user keeps changing as well as time evolves, the bandwidth reservation should be re-calculated periodically.

4.2.3 Integrated Bandwidth Management Process

In this section a detailed description of the proposed integrated Call Admission Control and resource reconfiguration process to be used for new and handoff calls, is provided. For simplicity in the presentation of the algorithm operation and without loss of generality, we assume there are two classes of traffic in the network and the QoS requirements are reflected by the corresponding effective bandwidth. Furthermore for simplicity in the description of the bandwidth reconfiguration algorithm operation, the terms bandwidth and effective bandwidth are used interchangeably.

- Class 1 (higher priority): The desired bandwidth for this kind of traffic is BW_1^u. If class 1 traffic cannot obtain the desired bandwidth, it may have the option to continue at a lower bandwidth requirement BW_1^l. For instance this can be achieved by adjusting the coding rate so that the video/audio quality is still acceptable (i.e. real-time traffic).

- Class 2 (lower priority): The desired bandwidth for this kind of traffic is BW_2^u and there are no strict QoS requirements. However, some flexible QoS requirements are defined for such a service. The user could specify a set of acceptable QoS levels that correspond to bandwidth requirements that range from a lower bound bandwidth requirement BW_2^l to a maximum bandwidth requirement BW_2^u, and expect a QoS varying in the specified range (i.e. non-realtime traffic).

Throughout the rest of this chapter we use the following notations:

- BW_{total}: total cell capacity;

- BW_1^{used}: total bandwidth used by all the calls that belong to class 1 in a cell;

- BW_2^{used}: total bandwidth used by all the calls that belong to class 2 in a cell;

- $BW_2^{minused}$: minimum bandwidth that should be reserved for current class 2 traffic being served in each cell, to meet the minimum bandwidth requirements for class 2

traffic. $BW_2^{minused}$ can be calculated as: $BW_2^{minused} = N_{class2} * BW_2^l$, where N_{class2} is the number of class 2 calls in the cell;

- BW_1^{res}: bandwidth reservation request in a cell for class 1 traffic (for handoff purposes) from all neighbor cells. BW_1^{res} can be calculated using equation (4) in [34] for only class 1 traffic, as follows:

$$BW_1^{res} = D \sum_{i \in N_j} \sum_{k \in S'_{i,1}} BW_1^u P_k^1 \tag{4.5}$$

where P_k^1 is the handoff probability of mobile user k belonging to class 1 traffic and $S'_{i,1}$ is the set of those ongoing class 1 calls which are currently in cell i and are going to handoff to cell j;

- BW_2^{res}: bandwidth reservation request in a cell for class 2 traffic (for handoff purposes) from all neighbor cells. BW_2^{res} can be calculated using equation (4) in [34] for only class 2 traffic, as follows:

$$BW_2^{res} = D \sum_{i \in N_j} \sum_{k \in S'_{i,2}} BW_2^u P_k^2 \tag{4.6}$$

where P_k^2 is the handoff probability of mobile user k belonging to class 2 traffic and $S'_{i,2}$ is the set of those ongoing class 2 calls which are currently in cell i and are going to handoff to cell j.

In the following we describe conceptually how does the proposed scheme operate, while later in this section we provide a pseudo-code that gives a detailed description of the process and the corresponding bandwidth allocated to each user in every case.

For a new connection the proposed scheme works as follows. For a class 1 new call the scheme first attempts to allocate the desired amount of bandwidth BW_1^u, if this is available, in the cell that the call is generated. If this is not available then the scheme tries to accept the new class 1 call at a slightly degraded quality (i.e. at bandwidth BW_1^l), however

still acceptable to the user. If there is sufficient available bandwidth to do so, then the call is accepted and the corresponding bandwidth is allocated. Otherwise the call is rejected. It should be noted here that the available bandwidth is calculated based on the total bandwidth capacity, the bandwidth currently used by active calls in the current cell and the bandwidth that has been reserved in the current cell for handoff purposes. The bandwidth reservation at each step is performed according to the PMBBR scheme as described above in relations (4.5) and (4.6), for class 1 and class 2 traffic respectively. It should be also pointed out here that the bandwidth to be reserved in a cell represents the collective effect of the influence that the class 1 (class 2) traffic from neighboring cells exerts on the cell under consideration, and does not refer specifically to individual users. Therefore this bandwidth is available to be used for all the calls that belong to the corresponding class as they may move into the target cell. A detailed calculation of the available bandwidth for each subcase, based on the notation introduced earlier, can be found in the pseudo-code of this procedure that follows later in this section.

For a class 2 new call the scheme first attempts to allocate the desired amount of bandwidth BW_2^u, if this is available, in the cell that the call is generated. If this is not available then the call is rejected. The reason that in such case we do not accept the new class 2 call at degraded quality where less bandwidth is required although such bandwidth could be available at the time of the call generation, is because we do not want to overload the system with a large number of class 2 calls supported at the lowest acceptable bandwidth (e.g. BW_2^l). Therefore we want to keep some bandwidth available for handoff calls (for both class 1 and class 2) and we also want to be able to borrow some bandwidth from ongoing class 2 calls and allocate that to class 1 and class 2 handoff calls. If we had accepted a large number of class 2 calls in the system at the minimum required bandwidth BW_2^l, then most of the bandwidth would have been allocated to these calls while at the same time there would not be sufficient resources to be borrowed from to support other handoff calls.

For handoff connections the proposed scheme works as follows. For a class 1 handoff call it first attempts to allocate the desired amount of bandwidth BW_1^u, if this is available, in the cell that the handoff call is moving into. If this is not available then the scheme tries to continue supporting the handoff class 1 call at slightly degraded quality (e.g. allocate bandwidth BW_1^l), however still acceptable to the user (according to the pre-specified characteristics and requirements of a class 1 user). If there is sufficient available bandwidth to do so, then the handoff call is accepted in the handoff cell and the corresponding bandwidth is allocated to it. Since this is a class 1 handoff call the system gives higher priority to this call and tries to accommodate it, even if some bandwidth reconfiguration is required in order to do so. In this case the bandwidth reconfiguration refers to class 2 calls that are currently supported in this cell. The system will try, if needed, to borrow bandwidth from current class 2 calls in this cell, that are supported with bandwidth higher than their minimum requirement. The detailed procedure of the bandwidth reconfiguration as well as the selection of the class 2 users to participate in this process, so that the overhead involved in such process is minimized, is described later in this section. In the case that there is not enough bandwidth (even if reconfiguration could take place) to support the handoff call then this call is dropped.

For a class 2 handoff call the proposed scheme first attempts to allocate the desired amount of bandwidth BW_2^u, if this is available, in the cell that the handoff call is moving into. If this is not available then the scheme tries to continue supporting the handoff class 2 call at some lower bandwidth within the user pre-specified range (i.e. between BW_2^l and BW_2^u). At this step the system attempts to allocate to the handoff class 2 call the maximum available bandwidth (within its pre-specified range) without involving the bandwidth reconfiguration process. As a last attempt to accept this handoff call the scheme tries to allocate the minimum required bandwidth BW_1^l to this call by implementing the bandwidth reconfiguration process. Again, as can be observed by the order of the above steps the bandwidth reconfiguration process is invoked as a last resort in an attempt to

minimize the overhead associated with this process and minimize the number of other class 2 calls impacted. Finally in the case that there is not enough bandwidth (even if reconfiguration could take place) to support the handoff call then this call is dropped.

The following pseudo-code provides a detailed description of the proposed strategy, as it should be implemented at each base station following the principles and notations provided above, and also describes the corresponding bandwidth allocation to each user in every case:

IF new call-arrival **THEN**

IF class 1 traffic **THEN**

IF $BW_1^u \leq BW_{total} - BW_1^{used} - BW_2^{used} - BW_1^{res} - BW_2^{res}$ **THEN**

accept connection and allocate BW_1^u to the new call;

ELSEIF $BW_1^l \leq BW_{total} - BW_1^{used} - BW_2^{used} - BW_1^{res} - BW_2^{res}$ **THEN**

accept connection and allocate BW_1^l to the new call;

ELSE /* not enough bandwidth */

reject connection;

ELSE /* class 2 traffic */

IF $BW_2^u \leq BW_{total} - BW_1^{used} - BW_2^{used} - BW_1^{res} - BW_2^{res}$ **THEN**

accept connection and allocate BW_2^u to the new call;

ELSE /* not enough bandwidth */

reject connection;

ELSE /* handoff call */

IF class 1 traffic **THEN**

IF $BW_1^u \leq BW_{total} - BW_1^{used} - BW_2^{used}$ **THEN**

accept connection and allocate BW_1^u to the handoff call;

ELSEIF $BW_1^l \leq BW_{total} - BW_1^{used} - BW_2^{minused}$ **THEN**

accept connection and allocate BW_1^l to the handoff call;

start bandwidth reconfiguration procedure if necessary (discussed later in this subsection);

ELSE /* not enough bandwidth */

drop handoff call;

ELSE /* class 2 traffic */

IF $BW_2^u \leq BW_{total} - BW_1^{used} - BW_2^{used} - BW_1^{res}$ **THEN**

accept connection and allocate BW_2^u to the handoff call;

ELSEIF $BW_2^l \leq BW_{total} - BW_1^{used} - BW_2^{used} - BW_1^{res}$ **THEN**

accept connection and allocate $BW_2^{available}$ to the handoff call;

$(BW_2^{available} = BW_{total} - BW_1^{used} - BW_2^{used} - BW_1^{res})$

ELSEIF $BW_2^l \leq BWtotal - BW_1^{used} - BW_2^{minused} - BW_1^{res}$ **THEN**

accept connection and allocate BW_2^l to the handoff call;

start bandwidth reconfiguration procedure (discussed later in this subsection);

ELSE /* not enough bandwidth */

drop handoff call;

As mentioned above, in some cases handoff traffic has to borrow bandwidth from current class 2 traffic in a cell. Therefore reconfiguration of bandwidth may be required. In the following we describe an approach which reduces the signaling overhead in cases where bandwidth reconfiguration is required. In order to minimize the effect of bandwidth reconfiguration on current class 2 traffic in the cell, we collect bandwidth from class 2 traffic that has the higher probability to handoff to neighbor cells in the near future. In each cell (at the base station), there is a pool used by the bandwidth reconfiguration procedure. Each time a class 2 traffic request is accepted in the cell with bandwidth more than BW_2^l, it will enter this pool. Whenever bandwidth is needed from current class 2 traffic, the traffic with the lowest predicted handoff time [34] in the pool will decrease its bandwidth until the lower bound BW_2^l and then be deleted from the pool. In this way, minimum number

of class 2 customers are involved into the bandwidth reconfiguration procedure so that signaling overhead is minimized.

The above CAC and resource reconfiguration scheme, always attempts to provide the required resources to meet the service quality of class 1 calls that present more strict QoS requirements, whether these are new call attempts or handoff calls. However at the same time, non-realtime traffic (class 2 traffic) that has been accepted into the wireless networks can maintain high successful handoff rate.

4.3 Management Paradigm Evaluation for the Comprehensive Resource Management Scheme for Wireless Networks

As explain before, and depicted in figure 4.1, the comprehensive resource management system has two sub-systems: bandwidth reservation subsystem and bandwidth reconfiguration/call admission control subsystem.

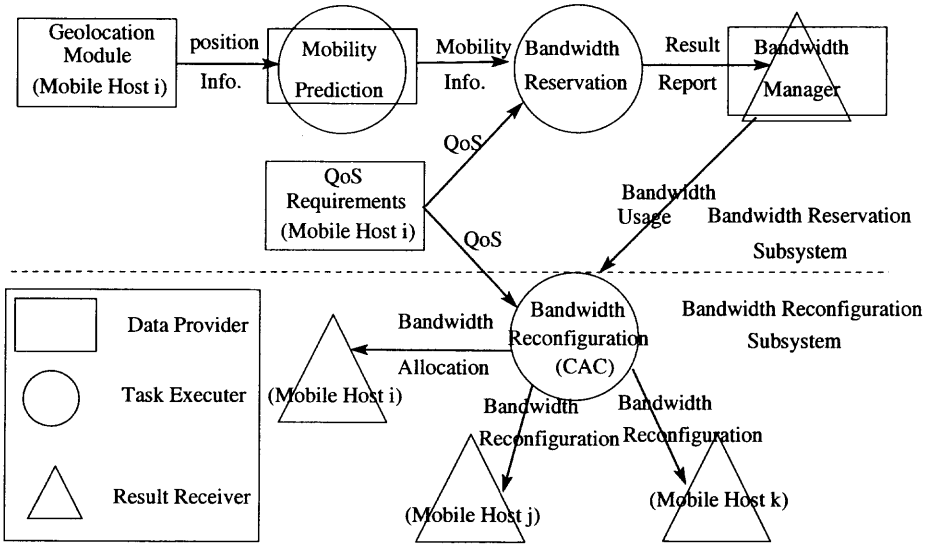


Figure 4.1 Information exchanges in the comprehensive resource management scheme

The various entities involved in the bandwidth reservation subsystem, and the corresponding functions, are as follows:

- Geolocation Module. This module is located at every mobile host and continuously detects the real-time position information of the mobile host by the means of GPS or other geolocation technology. This module is a Data Provider in the management activity.

- Mobility Prediction Module. This module owns the mobility information processing algorithm and can predict the future movement of the mobile host. The prediction results will be used by the Bandwidth Reservation module to calculate the bandwidth needs to be reserved. The Mobility Prediction Module is both a Task Executer and a Data Provider.

- QoS requirements Module. This module is located at every mobile host and can provide information about the Quality of Service requirements of current applications running in the mobile host. These requirements are necessary in order to calculate the bandwidth reservation. This is a Data Provider Module.

- Bandwidth Reservation Module. This module uses data from the Mobility Prediction Module and QoS requirement parameters to calculate the bandwidth needed for the possible handoff of the mobile host. This module is a Task Executer.

- Bandwidth Manager. This module is associated with the base station in order to get the report from Bandwidth Reservation Module. Then, it summarizes all the bandwidth reservation requests from the various mobile hosts within the cell it covers, and sends the comprehensive bandwidth reservation requests to other Bandwidth Managers in the neighboring cells. This Module is a Result Receiver.

The various entities involved in the bandwidth reconfiguration/call admission control subsystem, and the corresponding functions, are as follows:

- Bandwidth Configuration Module. This is the Task Executer who carries out the flexible bandwidth reconfiguration algorithm during the CAC process, as described in the previous section.

- QoS requirements Module. This is the same module listed in the above subsystem. In the bandwidth reconfiguration subsystem, this modules also acts as a Data Provider to provide QoS information to the base station in the Call Admission Control process.

- Bandwidth Manager. This module is the same module listed above. It can provide the bandwidth usage information to the Bandwidth Configuration Module.

- Mobile Hosts. Mobile Hosts are the Result Receivers in the CAC process: during the CAC procedure, the mobile host that requests to be accepted to the system will be allocated some bandwidth (if it is accepted), and the mobile hosts that are involved into the bandwidth reconfiguration process need to reconfigure their bandwidth usage according to the results from the Bandwidth Configuration Module.

Among the entities listed above, Data Providers and Result Receivers are associated with the corresponding network elements and can not move. On the other hand the Task Executers - the Mobility Prediction Module, the Bandwidth Reservation Module and the Bandwidth Reconfiguration module - could be flexible to associate with different network elements in the overall system. It should be noted that in essence, the bandwidth configuration scheme is a base station centralized scheme while the advanced bandwidth reservation scheme, as described above, is a decentralized scheme. The bandwidth reservation for each mobile host can be calculated separately by each mobile host, and it is only the bandwidth manager at the base station at the final calculation step, that integrates all these individual bandwidth reservations. On the other hand, for the bandwidth reconfiguration scheme, once the bandwidth reconfiguration module receive the CAC request from a mobile host, it needs to check the bandwidth usage information from the Bandwidth Manager at the base station, and run the corresponding bandwidth

reconfiguration scheme, in order to make the bandwidth re-configuration decisions. Finally, the CAC decision is sent to the mobile host that initiated the request, while the bandwidth re-configuration decisions are sent to the mobile hosts involved into the bandwidth re-configuration process.

In the next section, based on the model and framework described in chapter 3, we gain some insight about the performance of the above mentioned system under the Client-Server and the Mobile Agent paradigms. Following similar notations as the ones introduced in 3, for demonstration purposes, we calculate the following performance measures: a) Management traffic generated around the base station S_{BS}; b) Management traffic S_{Vi} around a mobile host i; c) Computation instructions executed by the base station (I_{BS})and by each mobile host i (I_{Vi}), in order to measure the computational capability used in every network device; c) Time consumed by the whole bandwidth reservation calculation procedure in order to evaluate the reaction speed of the management scheme, which is a critical parameter especially for real-time network management tasks.

4.3.1 Performance Evaluation in Client-Server Paradigm

Figure 4.2 depicts the interactions among the various entities in the Client-Server paradigm. As shown in this figure, in order to predict the future mobility of a mobile host, the Mobility Prediction module at the base station needs to query the Geolocation module at the mobile host over the wireless link. For a 2-order autoregressive prediction, three historical position information are needed. In CS approach, the management primitives are often low level and fixed, and no semantic compression is allowed to be performed [25]. So three historical position information need three query and reply interactions. Then the Bandwidth Reservation module at the base station needs the QoS parameters from the mobile host. Finally the calculation result is reported to the Bandwidth Manager at the base station. In the following let us assume that there are total Q mobile hosts in the cell covered by a base station. Based on the interactions in Client-Server paradigm, in the following, we

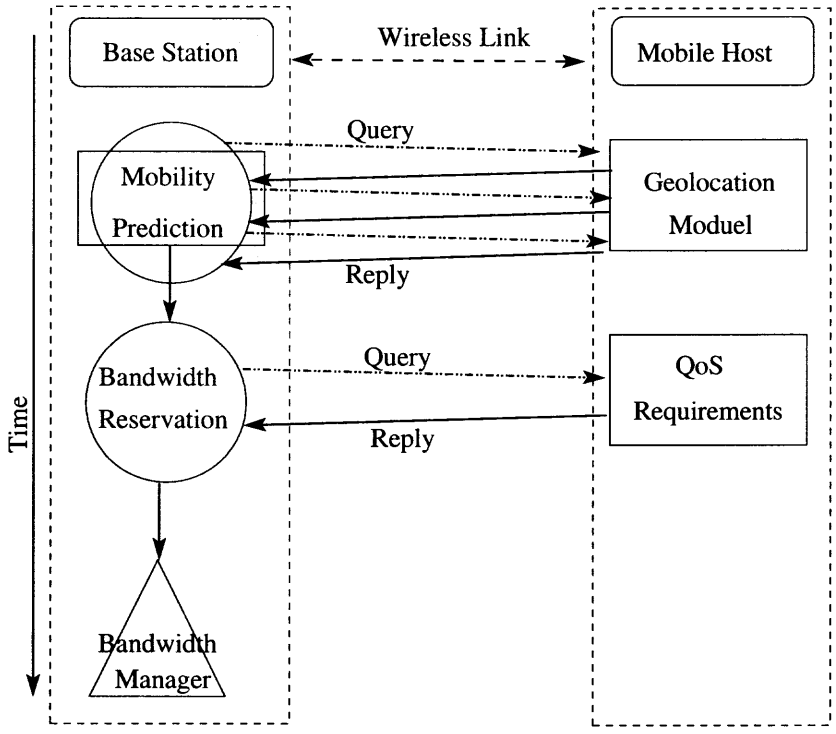


Figure 4.2 Bandwidth Reservation subsystem in CS paradigm

calculate the traffic generated around the base station and mobile hosts, the computational load of each network device and time used for the management task.

Let S_q^1 and S_q^2 represent the size of the queries (in bits) to obtain the mobility and QoS information. Similarly let S_d^1 and S_d^2 represent the size of the corresponding replies (in bits). Then the traffic generated around the base station is given by:

$$S_{BS}^{CS} = Q * \{3 * [(S_q^1 + S_d^1)] + (S_q^2 + S_d^2)\} \tag{4.7}$$

Similarly, the traffic generated around any mobile host i, S_{Vi}^{CS}, is equal to $1/Q$ of the corresponding traffic around the base station.

Let us also assume that the mobility prediction and the bandwidth reservation algorithms execute I_1 and I_2 computational instructions respectively. Then the instructions executed at the base station can be calculated as:

$$I_{BS}^{CS} = Q * (I_1 + I_2) \tag{4.8}$$

Since no management algorithm is executed at the mobile host we have that:

$$I_{Vi}^{CS} = 0 \tag{4.9}$$

The time consumed by the bandwidth manager to obtain the bandwidth reservation values for all the Q mobile hosts, includes the time used for data accessing over the wireless link (requests and replies), the time t_{MIB} used for data accessing from MIB (assume that all data are organized in MIBs in the network devices) and the time t_{alg}^{bs} used to execute the algorithm (mobility prediction and bandwidth reservation) at the base station. Therefore we have:

$$T_{total}^{CS} = \frac{S_{Vi}^{CS}}{Bw} + 8 * t_d + t_{MIB} + t_{alg}^{bs} \tag{4.10}$$

where Bw is the bandwidth of the wireless link and eight wireless link propagation delays t_d are included, because there are total four query-reply interactions over the wireless link in the Client-Server paradigm (three mobility information queries and one QoS information query).

4.3.2 Performance Evaluation in Mobile Agent Paradigm

In the Mobile Agent paradigm, instead of moving the data to the Task Executers at the base station as in the Client-Server paradigm, Task Executers (Mobility Prediction module and Bandwidth Reservation module) are transported to the mobile host, as Mobile Agents where the Data Providers (Geolocation Module and QoS parameters) are located. We should note here that, once the agents are embedded (downloaded) in the mobile station, unless a new resource management strategy is needed (when a management system is

upgraded or the mobile stations travels to a new foreign domain), we do not need to transport the code of agent again.

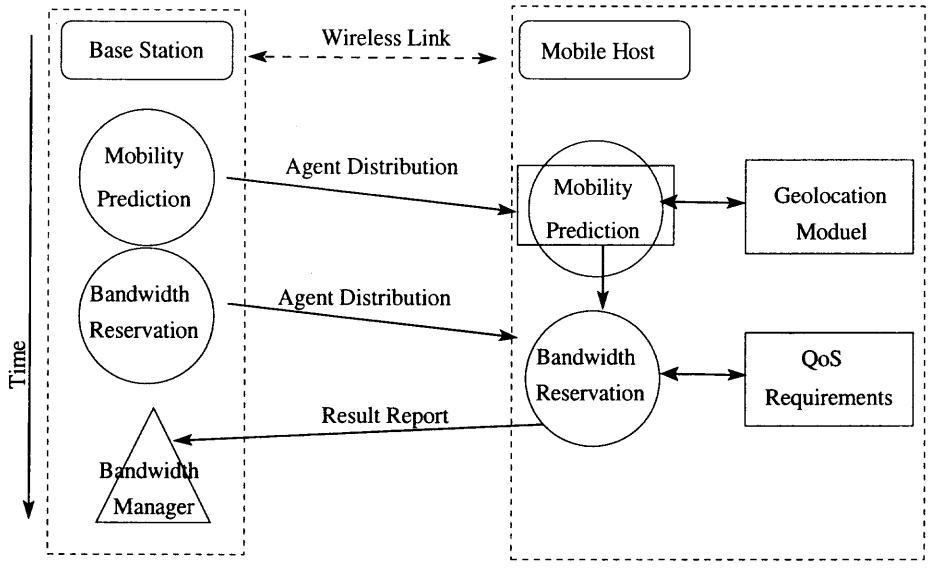


Figure 4.3 Bandwidth Reservation subsystem in MA paradigm

As shown in figure Figure 4.3, the previous remote data accessing over the wireless link becomes local data accessing within the mobile host. Therefore, after the initialization phase of the agent download, the total management traffic generated around the base station is due to the result reporting process can be calculated as:

$$S_{BS}^{MA} = Q * S_r \tag{4.11}$$

where S_r is the size of the bandwidth reservation result. Similarly, the traffic around any mobile host i, S_{Vi}^{MA}, is equal to $1/Q$ of the traffic around the base station.

Since in this scenario all the management algorithms are executed at the mobile hosts we have:

$$I_{BS}^{MA} = 0 \tag{4.12}$$

while the computational load at any mobile host i is:

$$I_{Vi}^{MA} = (I_1 + I_2) \tag{4.13}$$

It should be noted that in the MA paradigm, the bandwidth reservation computation could be done in parallel by Q mobile agents at the Q mobile hosts. The time consumed to inform the bandwidth manager about all the Q bandwidth reservation results in the MA paradigm includes the data accessing time t_{MIB}, the algorithm execution time t_{alg}^{mh}, and the result reporting time. Therefore:

$$T_{total}^{MA} = \frac{S_r}{Bw} + t_d + t_{MIB} + t_{alg}^{mh} \tag{4.14}$$

Please note that here only one wireless link propagation delay t_d is included, because there is only one result reporting interaction over the wireless link in the Mobile Agent paradigm.

4.3.3 A Comparative Example

In the following we obtain some numerical results regarding the performances of the CS and MA paradigms, by applying the relations developed in the previous sections, in a specific example. Let us assume that all the queries (S_q^1 and S_q^2) in the Client-Server paradigm have the same size of 32 bits. The size of the mobility information (x,y coordination information, moving speed and direction) is assumed to be 128 bits and the corresponding size of the QoS information (bandwidth requirement) is 32 bits. In this case, based on the relation developed in the previous section we have: $S_{BS}^{CS} = Q * 3 * [(S_q^1 + S_d^1)] + (S_q^2 + S_d^2) = Q * [3 * (32 + 128) + (32 + 32)] = 224 * Q$ bits, and $S_{Vi}^{CS} = 224$ bits for one bandwidth reservation calculation. Similarly in the MA paradigm, $S_{BS}^{MA} = Q * S_r = 32 * Qbits$, where S_r is the bandwidth reservation result which has the size of 32 bits. Furthermore $S_{Vi}^{MA} = 32$ bits for one bandwidth reservation calculation. Comparing the above numerical results, we can easily observe that the traffic generated around any network device in CS paradigm is seven times larger than the traffic in the MA paradigm. Similarly the overall traffic generated in the network in CS paradigm is also seven times the one generated in the MA paradigm.

Furthermore from relations (4.8), (4.9), (4.12) and (4.13) we can also observe that in the CS paradigm nearly all the computational load is distributed on the base station side, while in the MA paradigm, the computational load is evenly distributed among all the mobile hosts. As a result, especially in a heavily loaded system, with the increase of the number of mobile hosts within the coverage area of a base station, the bandwidth and computational capacity of the base station under the CS paradigm could become potential bottlenecks, which would affect significantly the performance and scalability of the overall management system.

As mentioned before, the total time consumed by a management task is also one of the key performance parameters, especially for the realtime management tasks. Although in the CS paradigm the base station has to handle Q algorithm executions in the time of t_{alg}^{cs}, generally the base station has more computational power and capacity than the individual mobile hosts. Therefore in the following we ignore the time difference between the t_{alg}^{cs} and t_{alg}^{mh}. Comparing the time consumed by one bandwidth reservation task in CS and MA paradigms based on relations (4.10) and (4.14), the corresponding time difference is: $T_{total}^{CS} - T_{total}^{MA} = \frac{S_{Vi}^{CS} - S_r}{Bw} + 7 * t_d = \frac{192bits}{Bw} + 7 * t_d$. For instance, given a wireless link with bandwidth of 128k bps and propagation delay of 1ms, the above difference could be 8.5ms. In a more severe wireless networking environment this difference would be even larger. In conclusion the CS paradigm is more sensitive to the variance of the wireless bandwidth and the propagation delay, compared to the MA paradigm.

Therefore, in order to achieve higher system scalability and faster management reaction speed, the MA based management paradigm is more suitable. At the same time, under this paradigm the computational load is evenly distributed among all the mobile hosts and the overall distributed nature improves the reliability and scalability of the system. In addition, the flexibility of the MA paradigm makes it possible that any special strategy for bandwidth reservation or special mobility prediction algorithm can be easily encapsulated

in the mobile agent and be implemented in the whole system, without any significant system upgrade.

4.4 Framework of Mobile Agent Based Geolocation and Resource Management System

In the following we describe a distributed management system based on mobile agent technology which can be used as the framework for the implementation and deployment of the integrated position-assisted handoff and bandwidth management scheme described above. As shown in figure 4.4, position-assisted handoff bandwidth reservation algorithm requires cooperation among various agents that carry out different tasks. The agents used in the proposed system and their corresponding roles are:

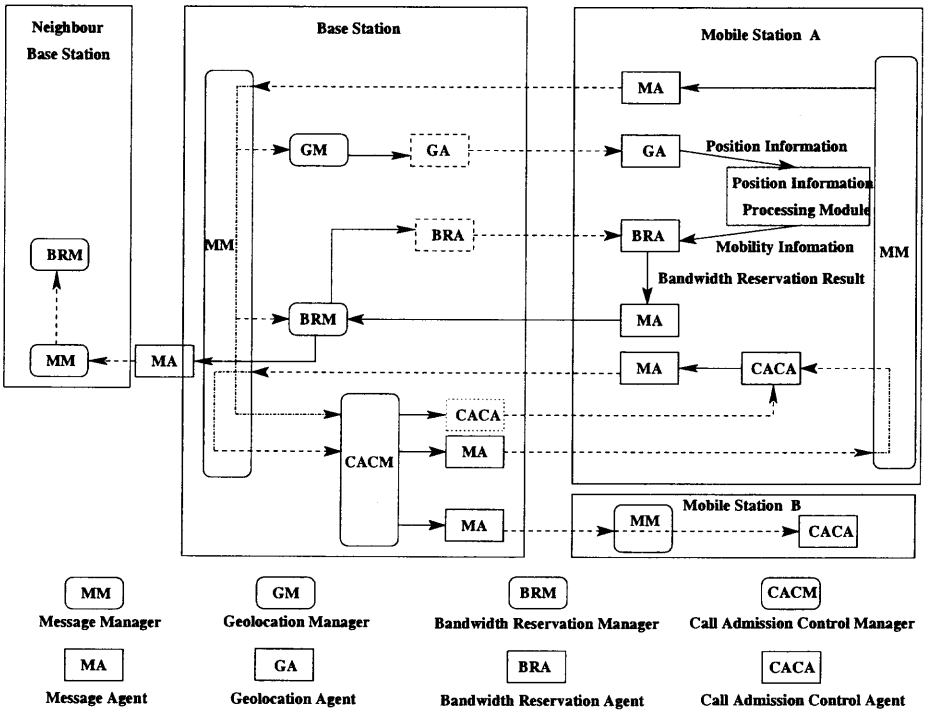


Figure 4.4 Mobile Agents used in resource management

- **Message Agent:** Message Agents (MAs) are used to exchange information and management data among agents and managers. They are created and received by

Message Managers (MMs). MM also could forward messages contained by MAs to the corresponding managers or agents in the same network element.

- **Geolocation Agent:** Geolocation Agents (GAs) are created by **Geolocation Managers** (GM) in base stations and are sent to mobile stations. They contain the signal processing algorithm for signal measurement and the triangulation algorithm used in the current networks.

- **Bandwidth Reservation Agent:** Bandwidth Reservation Agents (BRAs) are created by **Bandwidth Reservation Managers** (BRMs) and are sent to mobile stations. They contain the bandwidth reservation algorithm used in the current networks.

- **Call Admission Control Agent:** Call Admission Control Agents (CACAs) are created by **Call Admission Control Managers** (CACMs) embedded in base stations and are sent to mobile stations. The CAC strategy can be flexibly deployed in this way. CACAs will collect customer requirement and send necessary information by MAs to CACMs. Our CAC algorithm is carried by CACMs.

When a mobile station is turned on, an MA is sent out to the base station containing initial information about the mobile station. MM in the base station relays the information in MA to CACM, GM and BRM. CACA will create a new CACA and send it back to the mobile station. CACA will collect user information about QoS requirement and sent it with call-in requirement back to CACM by MA. Our CAC algorithm embedded in CACM will make the decision of accepting the new call or not. Once the new call is accepted, GM and BRM also will create a new GA and a new BRA respectively, and send them back to the mobile station. Within the mobile station, GA can output the position of mobile station itself periodically and these position information are input into Position Information Processing Module. This module then provides mobility information about mobile stations to BRA. BRA can use mobility information to calculate the bandwidth required for handoff purposes. The bandwidth reservation is also sent to base station by

MA. Finally, BRM at base station will summarize the bandwidth needed by each of its neighbor cell and send each neighbor cell an MA containing the corresponding bandwidth reservation requirements. BRM at base station in the neighbor cell can use this result to perform the bandwidth reservation process. In the case of handoff, a handoff requirement and also the QoS requirement will be sent by MA to the CACM embedded in target base station. The pool used by bandwidth reconfiguration procedure is maintained by CACM. After bandwidth reallocation, CACM will send out message agents to mobile stations informing them about the reconfiguration results.

4.5 Performance Analysis

This section presents the performance analysis of the proposed scheme. Specifically, we compare the performance results of our proposed integrated strategy where both advanced bandwidth reservation (via the PMBBR scheme [34]) and QoS management (via the call admission control and resource reconfiguration scheme) are implemented, with the corresponding results of a conventional system where the fixed bandwidth reservation is implemented, in terms of achievable new call and handoff call blocking probabilities. Furthermore in order to gain some insight about the individual impact of the different components of the proposed integrated solution, we investigated the performance of a system where only the direction-based advanced bandwidth reservation (via the PMBBR scheme) is implemented, while the bandwidth reconfiguration based call admission control component is not implemented. In the following we first describe the model and assumptions used throughout our performance study, and then we present the corresponding results of the comparative study.

4.5.1 Model and Assumptions

The wireless network used throughout this study is composed of 37 cells, each of which has six neighboring cells. The cell radius is set to be 1000 meters. In order to approximate

the performance of a large cellular system the cells are wrapped around to eliminate the border effect. The arrival of new calls initiated in each cell forms a Poisson process with rate λ. The life time of each call is exponentially distributed with mean 240 seconds [45, 46]. Additional system and traffic parameters are summarized in table 4.1. It should be noted here that our proposed framework aims to suggest a general approach that supports the seamless operation and provides flexibility for the resource management in the next generation wireless networks that support multimedia services. The bandwidth values for the different classes of service were chosen for illustration purposes. New voice coding technologies and emerging data applications may bring different bandwidth requirements to the wireless networks, which however can be easily fit into our framework. Throughout our simulation study we assumed that the desired bandwidth requirement for class 1 (e.g. voice) users is 30 Kbps and for class 2 (data) users is 50 Kbps. These parameters are selected based on some current realistic systems and some other research efforts that have been reported in the literature on this topic [47]. For instance a GPRS terminal is able to download data at the speed up to 40-50Kbps and the voice data is sent at 22.8Kbps.

Parameter	Value	Description
BW_{total}	$1000Kbps$	Total Bandwidth Capacity of a cell
BW_1^u	$30Kbps$	Desired bandwidth requirement for class 1 users
BW_1^l	$25Kbps$	Lower bound bandwidth requirement for class 1 users
BW_2^u	$50Kbps$	Desired bandwidth requirement for class 2 users
BW_2^l	$5Kbps$	Lower bound bandwidth requirement for class 2 users

Table 4.1 Simulation parameters

The mobility model used throughout this study is as follows. When a new call is initiated, the corresponding MS is assigned with a random initial position inside the cell, a random moving direction and an initial moving speed which is chosen according to a uniform distribution in the interval $[0, V_{max}]$mile/hr. The speed (v) and direction (ϕ) are

updated every time interval Δt, according to the following model:

$$v_{new} = \begin{cases} \min\{\max[v_{old} + \Delta v, 0], V_{max}\} & \text{when } p \leq 0.9 \\ 0 & \text{otherwise} \end{cases} \quad (4.15)$$

$$\phi_{new} = \phi_{old} + \Delta\phi \quad (4.16)$$

where Δv models the acceleration/deceleration of the mobile user and is a uniformly distributed random variable over the interval $[-5mile/hr, 5mile/hr]$; $\Delta\phi$ characterizes the user's change in moving direction and is a uniformly distributed variable over the interval $[-\Delta\phi_{max}, \Delta\phi_{max}]$; p is a uniformly distributed random variable over the interval $[0, 1]$. The use of variable p allows us to simulate the situation where a mobile user may stop occasionally during the course of moving. We set the mobility parameters to be $V_{max} = 60mile/hr, \Delta\phi_{max} = \pi/4$ which correspond to highly directional, fast moving traffic (e.g. highway traffic). The mobility update interval (Δt) is chosen to be $10sec$ throughout this study.

4.5.2 Numerical Results

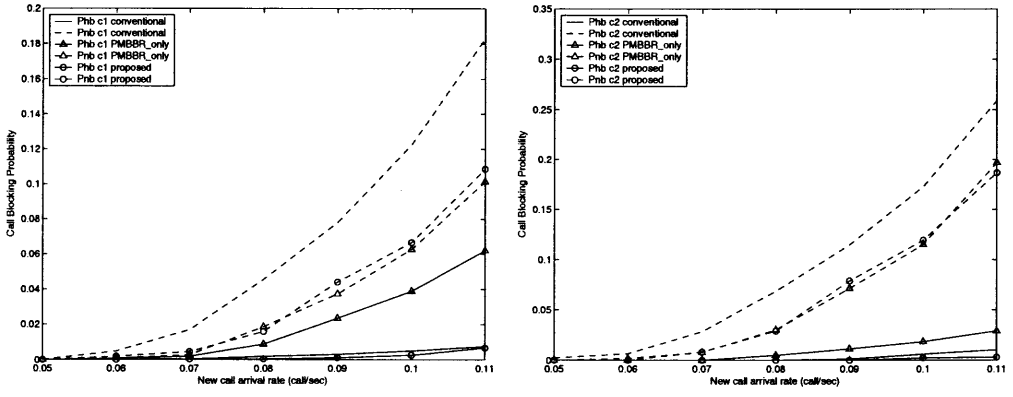
In the following we present the corresponding numerical results for two different test traffic scenarios regarding the composition of class 1 and class 2 traffic. Specifically, in scenario 1 10% of the new call attempts are class 2 calls, while in scenario 2 50% of the new calls are class 2 calls. Note that in the proposed CAC and resource reconfiguration scheme only class 2 calls can lend bandwidth to handoff calls, and therefore the number of ongoing class 2 calls in a cell will influence the performance of the system.

Figure 4.5 compares the handoff call and new call blocking probabilities of the proposed system with the corresponding results of the conventional system for different new call arrival rates (λ) under test traffic scenario 1, with users moving in the high speed pattern. In the legends of the figures, "c1" and "c2" stand for class 1 and class 2 users respectively, "proposed" indicates that the results are obtained under the

proposed integrated system where both position-assisted advanced bandwidth reservation and resource reconfiguration are implemented, "PMBBR_only" represents a system where only the advanced bandwidth reservation (via the PMBBR scheme) is implemented, while the the bandwidth reconfiguration based call admission control component is not implemented, and finally "conventional" corresponds to the case of a conventional system. The conventional system uses the fixed bandwidth reservation, where the reservation value represents a fixed percentage of the total capacity of the cell. In the following study the corresponding reservation value of the conventional system for class 1 users is $BW_1^{res} = 30Kpbs$ and for class 2 users is $BW_2^{res} = 50Kpbs$. The reservation values are selected based on experimentation with the objective of keeping similar handoff blocking probability for both our proposed system and the conventional system. The call admission control procedure for conventional system is the same as the one proposed in section 4.2 except that the bandwidth reconfiguration is not used. From the figure we observe that for the given parameters, the proposed system and the conventional system have similar handoff call blocking probabilities. However, the proposed system can significantly decrease the new call blocking probabilities for both user classes, which demonstrates that our proposed system can admit more users than the conventional system while still guarantee the same level of QoS for handoff calls.

The reason is two-fold: first, by using PMBBR algorithm, the bandwidth reservation is only made for those users that will request handoff in the near future and the reservation value can be dynamically adjusted according to the predicted user speed and direction. As can be seen by figure 4.5 compared with the "conventional" scheme, for both user classes, the "PMBBR_only" scheme achieves to decrease significantly the new call blocking probability at the cost of a slight increase in the handoff call blocking. Then by utilizing the bandwidth reconfiguration based CAC component the "proposed" scheme further improves the performance, by decreasing significantly the handoff call blocking probabilities with no impact on the new call blocking probabilities.This means that the

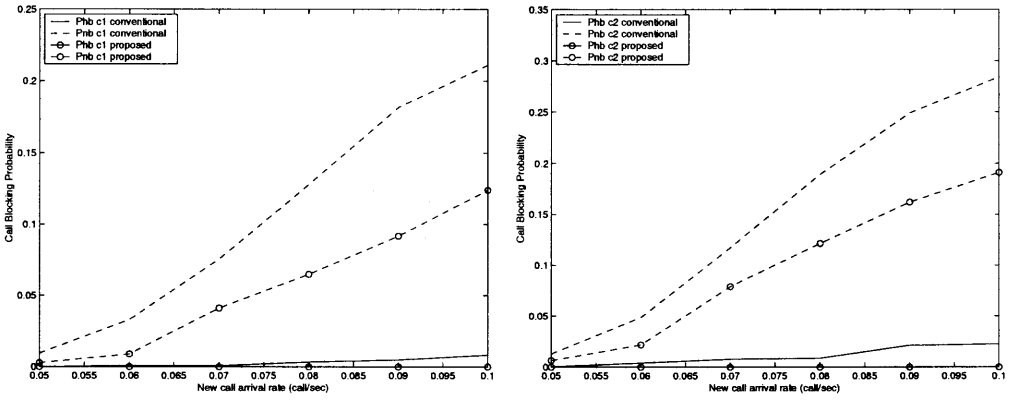
bandwidth reconfiguration component works complementary to the PMBBR scheme and eliminates any potential impact on handoff call blocking probability that may be introduced by the PMBBR scheme. This happens because the bandwidth reallocation procedure makes some ongoing class 2 users to reduce their current bandwidth usage to spare some bandwidth for the incoming handoff calls, which allows the system to achieve better handoff performance at relatively less reservation values. Both of them contribute to the reduction of the average reservation value without increasing the handoff failure probability. Less reservation value gives new calls better chance to be granted admission to the system. bandwidth reconfiguration component and the PMBBR scheme work together to improve the system resource utilization and guarantee seamless operation.



(a) (b)

Figure 4.5 Call blocking probabilities under test traffic scenario 1. (a) Call blocking probabilities for class 1 users. (b) Call blocking probabilities for class 2 users.

Figure 4.6 presents the corresponding numerical results for test traffic scenario 2 where 50% of the new calls belong to class 2 traffic. We observe that under this traffic configuration the proposed system can achieve to provide an even better performance improvement: the new call blocking probabilities are significantly decreased for both user classes, and the handoff call blocking probabilities are zero in the offered new call arrival rate range from 0.05 to 0.1. As a result, calls for both classes are never forced to termination



(a) (b)

Figure 4.6 Call blocking probabilities under test traffic scenario 2. (a) Call blocking probabilities for class 1 users. (b) Call blocking probabilities for class 2 users.

and the users obtain seamless connections once they are admitted into the system. Since each admitted class 2 call can spare some bandwidth (up to $45Kbps$ in our numerical study) to accommodate the handoff calls, if there are more class 2 users in service, the handoff call can always have sufficient bandwidth to borrow from when the reserved bandwidth is not enough to support it.

4.6 Conclusions

In this chapter we have proposed the integration of an advanced bandwidth reservation mechanism which facilitates the efficient and seamless operation of the handoff process, with a bandwidth reconfiguration based call admission control strategy that supports flexible QoS management. In cases of congestion and bandwidth shortage in a cell, the proposed integrated scheme allows traffic with strict QoS requirements to "borrow" some bandwidth from traffic with flexible QoS requirements. This bandwidth borrowing and reconfiguration process is designed such that the number of users involved in the process are minimized and therefore the associated overhead is minimized as well.

Furthermore a framework based on the technology of mobile agents, is introduced for the efficient implementation of the proposed integrated resource and QoS management. The performance evaluation study and corresponding numerical results demonstrated that our proposed integrated scheme can improve significantly the system resource utilization by alleviating the problem of bandwidth waste and can efficiently allocate the resources in the network to users with different QoS requirements. The mobile agent based architecture introduced in this chapter can speed up the bandwidth reservation and reconfiguration process, while at the same time the overall distributed nature can improve the reliability of the system. Several issues and design tradeoffs associated with the communications control overhead savings obtained by the proposed framework in comparison with a traditional resource management approach, the transport and execution of the mobile agent, as well as the required computational resources are also identified and discussed.

It should be noted here that in this chapter the use of the proposed mobile agent based resource management framework was demonstrated in detail for the case of bandwidth management. However the flexibility provided by the use of the agent technology simplifies and facilitates the consideration and management of other additional resources in wireless networks. Finally, any special strategy for resource management or special position measurement algorithm can be easily encapsulated in mobile agent and implemented in the whole system without any significant system upgrade.

CHAPTER 5

MOBILE AGENTS COOPERATION IN WIRELESS NETWORKS

5.1 Introduction

As mentioned in the Introduction, mobile agents in principle, can not only delegate the role of the managers in the conventional management paradigm, but can also cooperate with other agents to perform some functions and/or tasks more efficiently. The purpose of the cooperation among various mobile agents in network management process may be management function oriented and/or performance optimization oriented.

Currently, in the literature, the cooperation among mobile agents is mainly function oriented: two or more agents cooperate with each other to perform a specific task; without the cooperation of any of the agents involved in the task, the task can not be accomplished or executed correctly. For example, in [48], three kinds of agents - Network Monitoring Agents, Decision Agents and Action Agents cooperate with each other to perform the Intrusion Detection task in wireless Ad Hoc Networks. In [49] a cooperative multi-agent negotiation scheme for electronic commerce applications, based on mobile agents is discussed. In this scheme a buyer can send out a few Buyer Agents simultaneously to suppliers in order to speed up the negotiation process in the e-commerce environment. Buyer Agents keep exchanging quotation information during the negotiation process in order to achieve the best quotation from suppliers. In the resource management framework presented in chapter 4, in order to achieve more flexibility and reusability, two kinds of agents (Geolocation Agents and Bandwidth Reservation Agents) were created to perform the geolocation and bandwidth reservation calculation respectively. In order to perform the mobility prediction based bandwidth reservation task, the Bandwidth Reservation Agent has to cooperate with the Geolocation Agent to get the mobility information of the mobile host.

On the other hand, performance oriented agent cooperation mainly aims to improve the performance of the overall network management system. In [50], the use of multi-agents in a large scale distributed dynamic query system is demonstrated. Agent cooperation schemes are designed to minimize the number of control and data communication exchanges between the agents. The performance oriented agent cooperation should be as important as function oriented agent cooperation to perform the management task more efficient and/or reduce overhead of the management activity. In this chapter, based on the work introduced in chapter 4, a cooperation scheme for mobile agents involved in the resource management in the wireless network is proposed, in order to reduce the energy consumption in the management process.

5.1.1 Background Information

Over the past twenty years there has been steady progress towards making computers more mobile. This has been driven by both higher processor integration levels and the development of high-performance batteries. With the rise of packet-oriented wireless networks and powerful mobile terminals, various applications/services are under development such as location-aware navigation guides, financial applications, personal assistance services and entertainment [51][52]. For example nowadays, more than 50 percent of Japan's 60 million cell phone users have web-enabled phones and NTT DoCoMo's i-Mode service has 20 million subscribers [53]. Today's mobile terminals have the operation platforms to support more general applications/protocols such as Wireless Applications protocol (WAP) or i-Mode. Although it has been a trend for mobile devices to have enough computational power and suitable operation system to support various applications, all these enhanced features should be integrated and achieved with energy efficiency in all phases of the system design. While power consumption of the electronics and the display have been dropping dramatically forming a new energy-based Moore's Law [54], fundamental limitations of radio technology have prevented a similar trend in

wireless communications. It is estimated that in the near future nearly 80 percent of the power consumed by mobile computers will be due to communications [54]. As a result enhanced distributed resource management architectures [5][13][34] have been proposed in the literature that reduce the need for communication and control information exchange between the network management station (e.g. base station) and the corresponding network elements (e.g. mobile terminals), which in turn results in significant reduction of the wireless communications power consumption.

In conventional infrastructure-wireless networks in order to facilitate the efficient implementation of several management processes such as geolocation, bandwidth reservation, handoff, and call admission control, as well as support enhanced services such as location based services, a Mobile Host to Base Station (MH-to-BS) resource management information exchange paradigm has been adopted. In this paradigm all the resource/control related information updates (e.g. bandwidth required for the possible handoff to neighboring cells, position information, etc.) are sent directly and individually from each mobile node to the base station. In the following this mode is referred as direct reporting. However it is well known that in many cases multi-hop transmissions may result in significant power savings compared to a single-hop (equivalent) transmission [55]. Direct single-hop resource management information exchanges between the mobile terminals and the base stations could prove to be very costly in terms of traffic generated and energy consumed, especially in next generation wireless networks where a highly dense network is expected. Since in dense networks the system resources will be extremely scarce, it is critical to design efficient communication of control and management functions.

5.1.2 Motivation and Objective

In this chapter motivated by these observations we partially adopt the peer-to-peer communication concept of wireless ad-hoc networks, and we propose the use of a cooperative reporting scheme by allowing some mobile agents associated with mobile

terminals to form an ad-hoc cluster when they need to transmit resource management related information to the base stations (the Bandwidth Reservation Agent described in Chapter 4). In such an approach the network is subdivided into ad hoc subnetworks (clusters) where the agents of each cluster cooperate in order to perform the required management functions, reducing the overall wireless communication cost. All mobile agents (MA) in a cluster will forward their information to the cluster head agent (CHA), instead of each one of them individually sending its information to the base station. Then the CHA aggregates the information (including its own information) and transmits the complete report to the base station. In this kind of cooperative reporting scheme, the conventional individual direct report transmissions of the MAs to the base station are replaced by two-hop transmissions. The CHA acts only as resource management report aggregator to relay information to the base station. It differs from the base station concept in cellular networks, in that it does not need any special hardware, and in fact it is dynamically selected among the set of MAs. Such collective reporting results in significant reduction in the overall system energy consumption and the signaling traffic to the base station [56]. It should be noted here that in this chapter the proposed cooperative mechanism is adopted for the implementation of the reporting of the resource/management related control information only, while the transmission of the actual user data is still implemented as in conventional wireless networks.

Taking into account the combined advantages of cellular and ad hoc networks, several efforts have been reported in the literature, which have attempted to incorporate the concept of peer-to-peer communication into cellular networks therefore creating a new architecture for the future mobile networks [57] [58]. The majority of these works have mainly placed their emphasis on the following aspects: a) improve various system and service performance metrics such as capacity, throughput, channel usage, call blocking/dropping probability etc. (e.g. [59]); b) achieve traffic load balancing and interoperability in heterogeneous systems [60]; c) develop efficient routing techniques in multi-hop cellular

networks [61]. Our work, on the other hand, mainly aims to design, and demonstrate the use of cooperative methods that can be applied in such hybrid networking architectures to improve the energy consumption for routine management processes of mobile terminals, that need to be continuously and periodically implemented, in order to better meet the requirements of the applications of future wireless networks.

The rest of this chapter is organized as follows. In section 5.2 we outline some assumptions and notations that will be used throughout this chapter, and we provide some preliminary results on the conditions and the corresponding energy savings that can be achieved by the clustering multi-hop approach in wireless networks. Based on these observations in section 5.3 we provide a detailed description of the proposed distributed cooperative reporting scheme for resource management in wireless networks. The scheme consists mainly of two parts: the distributed clustering algorithm (Random Timeout Cluster Head Selection - RTCHS) and the query-reply-report information delivery interactions among the cluster head agent, member agents and the base stations. In section 5.4 the fairness of the proposed mechanism in selecting different nodes as cluster head agents is analyzed and discussed, and an enhanced algorithm, namely the Fair Variable Time Window RTCHS (FVTW-RTCHS) algorithm, is introduced in order to improve the fairness in consecutive cluster head selection cycles. This is achieved by introducing a new parameter, the Fairness Index (FI) which not only provides a measure of the fairness in selecting the CHA, but it is also used by the algorithm as a feedback from the previous selection cycles to appropriately adjust the probability that an agent is selected as CHA. Section 5.5 contains the performance evaluation of the proposed cooperative schemes along with some numerical results and discussions. Finally section 5.6 concludes the chapter.

5.2 Assumptions and Observations

Before we proceed with the description and analysis of the proposed cooperative scheme and the corresponding clustering operation, in this section we present some useful results on

the conditions and the corresponding energy savings that can be achieved by the clustering multi-hop approach in wireless networks. In order to gain some insight we Initially analyze the simplest case by considering one cluster with one CHA and one MEmber Agent (MEA). Then we expand our analysis and our results to the more general case where a cluster may support any number of MEAs.

Let us assume that at some certain time point, Q mobile agents associated with Q different terminals are going to report their information updates to the corresponding agent at the base station. The information reported by each mobile agent is assumed to be m bits long including the terminal's ID and management related data, such as position of the MT and bandwidth reservation information for possible handoff. We also assume that with each transmitted packet there is a fixed length transmission overhead of m' bits long. In the following for simplicity in the representation we consider that: $m' = km$ (generally $k < 1$). Furthermore, assuming that the transmission rate is fixed and that the signal strength decreases inversely proportional to d^x, where d denotes the distance between the transmitter and receiver, the energy e used to transmit z bits of information from a transmitter to a receiver d meters away is: $e = Azd^x$, where A is a power related constant. In the following assume $x = 4$ by the two-ray ground reflection model [62]. For the convenience of discussion, in the following of this chapter, assume each mobile terminal has a mobile agent associated with it who need to update information periodically to the base station (eg. the Bandwidth Reservation Agent in the last chapter) and the terms of mobile agent (MA) and the mobile terminal (MT) with which the mobile agent located may be used alternatively.

5.2.1 One Cluster Head Agent with One Member Agent

With reference to figure 5.1 let d be the distance between the base station and the mobile terminal where CHA is located (here denoted by terminal H), r_i the distance between MT i (where the mobile agent i is located) and the CHA, and θ_i the angle of MT i with

reference to the horizontal axis. Furthermore let us denote by $e^i_{j,f}$, the energy consumed by transmitter j to deliver to receiver f the information that was generated (and relayed to j) by node i. For notation consistency we denote by $e^j_{j,f}$ the energy consumed by transmitter j to deliver to f its own information in the direct reporting mechanism. In direct reporting scheme, terminals i and H transmit their information directly (separately) to the base station. Therefore the energy usage e_{dct} to deliver these messages is:

$$\begin{aligned} e_{dct} &= e^H_{H,BS} + e^i_{i,BS} \\ &= A(1+k)md^4 + A(1+k)m(\sqrt{(d - r_i \sin\theta_i)^2 + (r_i \cos\theta_i)^2})^4 \\ &= Amd^4(1+k)\{1 + [(1 - l_i \sin\theta_i)^2 + (l_i \cos\theta_i)^2]^2\} \end{aligned} \tag{5.1}$$

where $l_i = r_i/d$ represents the ratio of the distance between terminal i and the CHA to the distance between the CHA and the base station. In the cooperative reporting mode, the

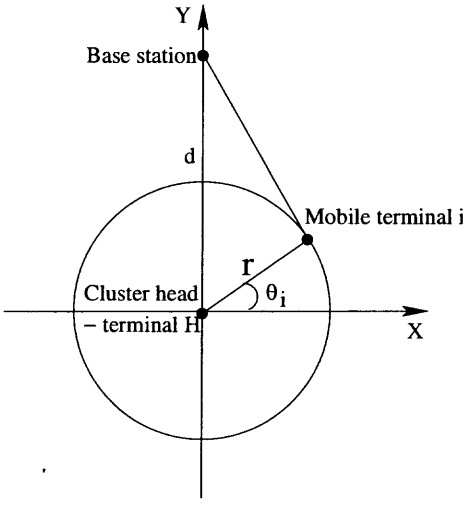


Figure 5.1 System geometry and energy usage in direct and cooperative mode

message from terminal i is delivered first to the CHA and the CHA delivers the combined message to the base station. In this case the corresponding energy usage e_{coop} is:

$$
\begin{aligned}
e_{coop} &= e^{i}_{i,H} + e^{i+H}_{H,BS} \\
&= A(1+k)mr_i^4 + A(2m+m')d^4 \\
&= Amd^4[(1+k)l_i^4 + 2 + k]
\end{aligned}
\tag{5.2}
$$

From relations (5.1) and (5.2) we can easily calculate the energy usage difference between the two different forms of resource management information updates (direct reporting and cooperative reporting):

$$
e_{dct} - e_{coop} = Amd^4\{(1+k)[(1-l_i\sin\theta_i)^2 + (l_i\cos\theta_i)^2]^2 - [(1+k)l_i^4 + 1]\} \tag{5.3}
$$

Based on this let us define by Δ^i_{energy} the corresponding energy difference factor as follows:

$$
\Delta^i_{energy} = (1+k)[(1-l_i\sin\theta_i)^2 + (l_i\cos\theta_i)^2]^2 - [(1+k)l_i^4 + 1] \tag{5.4}
$$

It can be easily seen that in this case the cooperative reporting approach can save energy compared to the direct reporting if Δ^i_{energy} remains positive. In figure 5.2 we present some numerical results of the energy difference factor Δ^i_{energy} as a function of parameter θ_i for the case where $k = 0$ (e.g. neglecting the transmission overhead). The different curves correspond to different values of parameter l_i (ratio of the distance between terminal i and the CHA to the distance between the CHA and the base station). From this figure we observe that the energy difference factor Δ^i_{energy} could be positive or negative depending on parameter θ_i, which represents the relative position of the MT to the CHA and the base station. We also observe that Δ^i_{energy} keeps positive approximately in the area $\{\theta_i|\pi < \theta_i < 2\pi\}$. If we assume that the MT is randomly and uniformly distributed around the CHA (i.e. θ_i is uniformly distributed in $[0, 2\pi]$) the average energy difference factor Δ^i_{energy} is obtained as follows:

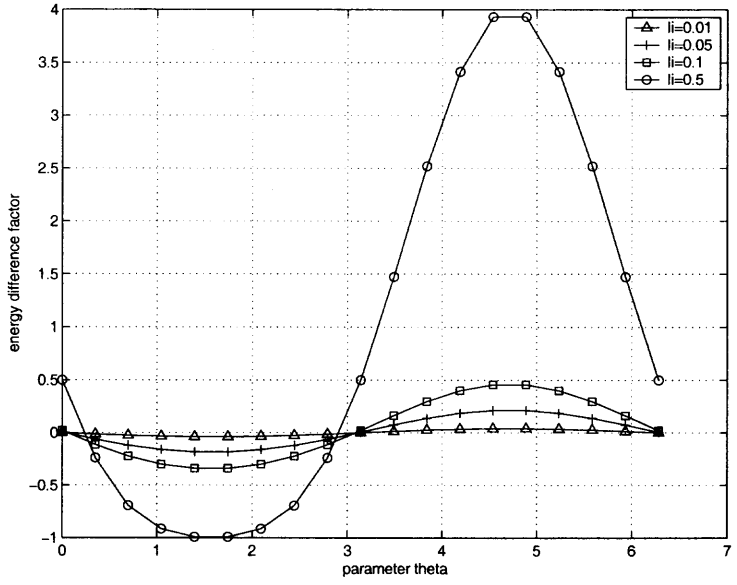


Figure 5.2 Energy Difference Factor vs. Parameter θ for Different Values of l_i

$$
\begin{aligned}
E[\Delta^i_{energy}] &= \int_0^{2\pi} \Delta^i_{energy} d\theta_i \qquad (5.5)\\
&= \int_0^{2\pi} \{(1+k)[(1-l_i \sin\theta_i)^2 + (l_i \cos\theta_i)^2]^2 - [(1+k)l_i^4 + 1]\} d\theta_i \\
&= 8(1+k)\pi l_i^2 + 2k\pi \\
&= (1+k)E[\Delta^i_{energy(k=0)}] + 2k\pi
\end{aligned}
$$

where

$$E[\Delta^i_{energy(k=0)}] = 8\pi l_i^2 \qquad (5.6)$$

Therefore based on equation (5.6) we can easily see that the expectation of Δ^i_{energy} for $k = 0$, $E[\Delta^i_{energy(k=0)}]$, is always positive and independent with the parameter l_i. In this case even though in principle the member terminal may not always save energy using the cooperative reporting scheme, if it is randomly and evenly distributed around the CHA, on the average the cooperative reporting mechanism is more energy efficient.

In the following we investigate the impact of parameter k, which is the ratio between the size of the overhead and the size of the message each MT should report to the base station, on the energy difference factor Δ^{i}_{energy}. Specifically figure 5.3 demonstrates how the energy difference factor changes as a function of parameter θ_i for different values of k, for the case where $l = 0.1$. From this figure we observe that as k increases the corresponding curves move upwards. That means the larger the overhead, the higher the chance for the cooperative reporting scheme to save energy. Especially, if $k = 0.5$, independent of the value of θ the cooperative reporting mechanism results in energy savings. This is due to the fact that for larger values of k more overhead is saved during the transmission from the CHA to the base station. Therefore if a group of MTs could form a cluster and deliver their information to the base station by the cooperative 2-hop mechanism significant energy savings may be achieved.

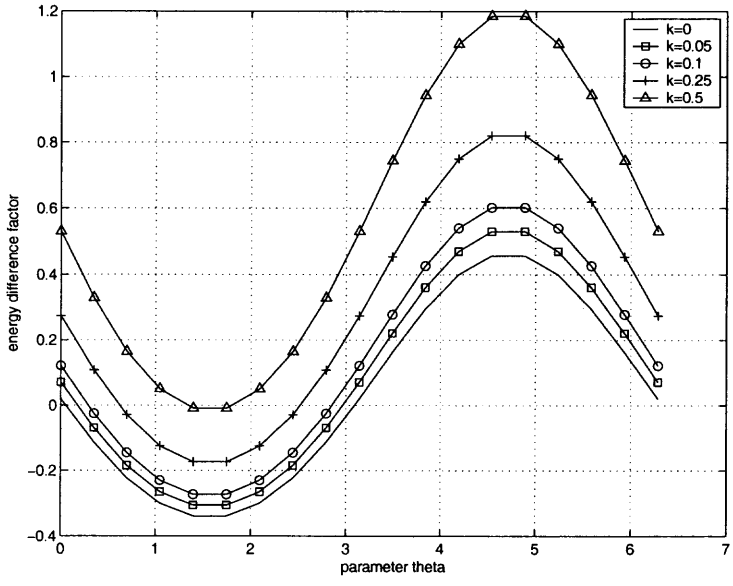


Figure 5.3 $\Delta_{energy(l=0.1)}$ vs. parameter θ for different values of k

5.2.2 Cluster with Q Mobile Agents

In this subsection we expand the above analysis for the case where Q terminals (Q agents) are supported in a cluster. In direct reporting mode, all Q terminals report their information directly to the base station with 1-hop radio transmission. In the following without loss of generality let us also assume that the agent at terminal Q acts as the CHA. Then in cooperative reporting mode, agents at terminals 1 to $Q-1$ deliver their information first to the CHA - terminal Q; then the CHA delivers the aggregated information (including its own information) to the base station. Following a similar approach as before the corresponding energy consumptions e_{dct} and e_{coop} can be obtained as follows:

$$\begin{aligned} e_{dct} &= \sum_{i=1}^{Q} e_{i,BS}^{i} \\ &= A(1+k)md^4 + \sum_{i=1}^{Q-1} A(1+k)m(\sqrt{(d-r_i\sin\theta_i)^2+(r_i\cos\theta_i)^2})^4 \\ &= Amd^4(1+k)\{1+\sum_{i=1}^{Q-1}[(1-l_i\sin\theta_i)^2+(l_i\cos\theta_i)^2]^2\} \end{aligned} \tag{5.7}$$

$$\begin{aligned} e_{coop} &= \sum_{i=1}^{Q-1} e_{i,Q}^{i} + e_{Q,BS}^{1+2...+Q} = \sum_{i=1}^{Q-1} A(1+k)mr_i^4 + A(Qm+m')d^4 \\ &= Amd^4[\sum_{i=1}^{Q-1}(1+k)l_i^4 + Q + k] \end{aligned} \tag{5.8}$$

The corresponding difference of the two energy consumptions is calculated as follows:

$$\begin{aligned} & e_{dct} - e_{coop} \\ = \; & Amd^4(1+k)\{1+\sum_{i=1}^{Q-1}[(1-l_i\sin\theta_i)^2+(l_i\cos\theta_i)^2]^2\} \\ & -Amd^4[\sum_{i=1}^{Q-1}(1+k)l_I^4 + Q + k] \\ = \; & Amd^4\sum_{i=1}^{Q-1}\{(1+k)[(1-l_i\sin\theta_i)^2+(l_i\cos\theta_i)^2]^2 - [(1+k)l_i^4+1]\} \\ = \; & Amd^4\sum_{i=1}^{Q-1}\Delta_{energy}^{i} \end{aligned} \tag{5.9}$$

The average energy usage difference for the whole cluster with Q terminals is obtained as:

$$\begin{aligned} E[e_{dct} - e_{coop}] &= E[Amd^4 \sum_{i=1}^{Q-1} \Delta^i_{energy}] = Amd^4 \sum_{i=1}^{Q-1} E[\Delta^i_{energy}] \\ &= Amd^4\{2k(Q-1)\pi + 8(1+k)\pi \sum_{i=1}^{Q-1} l_i^2\} \end{aligned} \tag{5.10}$$

Based on the latter expression we can easily see that $E[e_{dct} - e_{coop}]$ can always keep positive, even if $k = 0$, therefore resulting in energy savings of the cooperative mode over the direct reporting mode. From another point of view, the cooperative reporting scheme also achieves to reduce the control traffic to the base station, which in some cases may become a bottleneck due to the high traffic volume that needs to be handled by the base station. If all the Q mobile terminals send their information updates directly to the base station, the base station will receive total traffic of $Q(1+k)m$ bits, while under the cooperative mode, only $Qm + km$ bits are sent to the base station, which results in a difference of $(Q-1)km$ bits.

5.3 A Distributed Cooperation Scheme for Information Updating in Wireless Networks

In this section based on the observations and results presented in the previous sections, we provide a detailed description of the distributed cooperative reporting scheme for resource management in wireless networks. The scheme consists mainly of two parts: clustering algorithm and query-reply-report information delivery interactions among the cluster head, member agents and the base stations.

5.3.1 Clustering Scheme Design

In order to form the ad hoc cooperative reporting clusters before the information reporting, the CHAs should be selected. Then the CHAs could inform their possible member agents to collect the information and report the aggregated information to the base station.

It should be noted here that subnetworking by clustering in wireless networks has been studied in the literature in recent years (e.g., [63][64]). The clustering algorithms proposed in these works are mainly for the purpose of multihop routing for any two mobile terminals in wireless networks without the involvement of base stations (or access points), especially in ad hoc wireless networks. The expected advantages from the clustering include reuse of wireless resources, efficient routing management and power considerations. As a result, the algorithms proposed in the literature, consider various factors such as channel efficiency, route connectivity and performance optimization, and in many cases are quite complicated and present large overheads. In our work the main objective of clustering is the formation of independent groups with nodes close to each other in order to reduce the energy and communication cost involved in reporting the resource management related information to the base station. The algorithm should be executed in a distributed fashion and within a relatively short time, in order not to affect the management information updating process and reduce the overhead to form the ad-hoc clusters. To reduce the complexity of the clustering algorithm, one-hop clustering [65] approach is selected here, that is all member agents are only one hop away from the CHA.

Taking these factors into consideration the following distributed clustering algorithm is proposed:

- Each agent at the MT starts a CHA selection timer at time point A (see figure 5.4). This timer, which is randomly and uniformly distributed within a fixed time window starting at time point A and ending at time point B, will time out at time point C. Let us denote by U the next time point that resource management information updates are required by the base station. The ending time point B of the time window should be earlier than the pre-specified information update point U to allow the information collection by the CHA.

- If an agent times out before getting any query from other agents, it sends out a query message with controlled power that only terminals within a certain predefined query

range r can hear. The agent sending out the query becomes the CHA. The impact of the query range r (which defines the cluster size) on the overall performance of the proposed approach is discussed in detail in section 5.2.

- If an agent receives a query from a CHA, it cancels its own CHA selection timer and becomes a member to this CHA. The agent replies to the query with its updated information.

- Before the updating time point U, the CHA should receive all the replies within its query range and send the aggregated information to the base station.

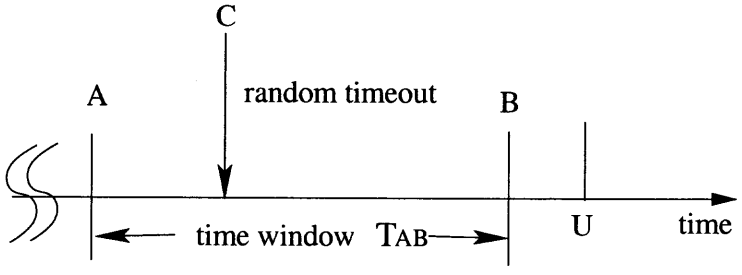


Figure 5.4 Time window of the cluster head selection algorithm

In the above scheme, the ad-hoc agent clusters are formed in a totally distributed manner. No central station is involved nor any information need to be exchanged among the individual terminals before the CHA is determined. Because the timeout point is randomly and uniformly distributed in a fixed time window T_{AB}, in principle every agent has the same probability to be selected as the CHA. We refer to this scheme as the Random Timeout Cluster Head Selection (RTCHS) algorithm. Later in the chapter an enhanced version of this scheme is designed in order to improve the fairness of the overall CHA selection process, especially in consecutive CHA selection cycles.

5.3.2 Query-Reply-Report Interaction

As mentioned before once a CHA is determined, the CHA broadcasts a query message to inform its possible cluster members. After hearing the query, the member agent sends all

the management related data, destined for the base station, to the CHA first. The specific query-reply-report interaction scheme assumed here is as follows. The CHA broadcasts its ID and its coordinates to the possible neighbor nodes (e.g. within a certain range r). Then the MEA that heard the query calculates the distance to the CHA and adjusts its transmission power to the required level to deliver its information update to the CHA. In certain time period, the CHA may collect all the replies from MEAs and then report the aggregated information to the corresponding agent at the base station.

5.4 Fairness Improvement of the Clustering Algorithm

5.4.1 Fairness Discussion of the RTCHS Scheme

The proposed RTCHS algorithm is a fair scheme for every agent in each independent CHA selection cycle, in the sense that all agents have the same probability to be selected as CHA. Specifically if Q agents are involved into the CHA selection procedure, since all the agents have the same CHA selection time window and the timeout point is uniformly distributed in the time window, they all have the same probability of $1/Q$ to be selected as CHA.

However the RTCHS algorithm is a memoryless algorithm in successive CHA selection processes (cycles). Let us consider the situation where the CHA selection may repeat N times during the residual time of MTs in a cell and the Q terminals are always close enough to form a cluster. Lets us denote by x the random variable that represents the times that an agent is selected as CHA throughout these N cycles. Random variable x follows binomial distribution with expected value $E[x] = N/Q$, and variance equal to $N(1/Q)(1 - 1/Q)$. This means that the variance may increase with the value of N - total CHA selection cycles. Therefore it is possible that some agents could keep to be selected as CHAs for a relatively long time. The direct effect is that these agents that act as CHAs for long time consume more energy and they may exhaust energy of the MTs they associated with much faster than other terminals. As a result the RTCHS scheme may introduce

unfairness during the various cycles of the CHA selection procedure. Similar unfairness may be introduced due to the mobility of the various MTs.

5.4.2 An Enhanced Fair CHA Selection Scheme

In order to improve the fairness of the RTCHS algorithm, we propose a Fair Variable Time Window (FVTW) enhanced RTCHS (FVTW-RTCHS) algorithm. To achieve that, for each mobile we introduce a new parameter, the Fairness Index (FI). The FI on one hand provides a measure of the fairness in selecting the CHA, while on the other hand is used by the algorithm as a feedback from the previous selection cycles to adjust the probability that an agent is selected as CHA in order to achieve fairness. The FI is updated as follows. Every time a mobile agent is selected as CHA, for each mobile agent it supports (e.g. for each member agent of its cluster) it receives one credit point (i.e. its FI increases by one unit). That is, the FI of an MT that is selected as CHA will increase in one cycle by as many units as the number of the members of its cluster. At the same time when a member agent is supported via some other CHA it looses one credit point (e.g. its FI is reduced by one unit). If FI equals to 0, the Mobile Agent is in a balanced fairness status; otherwise, the MT is in an unbalanced status.

In order to achieve fairness in an easy and efficient way we can use the FI of each mobile in a distributed and dynamic way to control the time window used for the CHA selection algorithm. Specifically, as shown in figure 5.4, let us suppose that in the RTCHS algorithm, the starting and ending time points of the fixed time window are A and B respectively ($B > A$). In FVTW algorithm, the starting and ending time points of the time window are denoted by a and b respectively, while before the cooperative reporting, all the agents are assumed to have $FI = 0$. According to the the proposed FVTW-RTCHS algorithm , each MT should execute the following steps:

1. Compute the starting and ending points a and b of the time window as:

$$a = \begin{cases} A, & \text{if } IF \leq 0, \\ A + \sum_{i=1}^{FI} (\frac{B-A}{w})^i & \text{if } IF > 0, \end{cases}$$

$$b = \begin{cases} B - \sum_{i=1}^{|FI|} (\frac{B-A}{w})^i, & \text{if } IF \leq 0, \\ B & \text{if } IF > 0, \end{cases}$$

2. Execute the RTCHS algorithm;

3. Update the Fairness Index as explained before;

In the algorithm above, w is the window shrinking factor (larger than 1). In this chapter, for demonstration only purposes and without loss of generality, we set $w = 2$.

In the following we present some properties related to the operation of the proposed FVTW-RTCHS algorithm, which demonstrate its capability to improve the CHA selection fairness and to guarantee some bounds on the Fairness Index, which can not be satisfied under the operation of the RTCHS algorithm as mentioned before.

Corollary 1: In a fixed cluster Φ with Q MTs (i.e. all nodes remain in the cluster during successive cycles of the cooperation process), the sum of the Fairness Indices of all the Q nodes is always equal to 0.

Corollary 2: When the FVTW-RTCHS algorithm is applied to select the CHA, terminals with the same FI have the same probability to be chosen as the CHA.

Property 1: A agent with positive Fairness Index will not be selected as the CHA if there is at least one terminal with negative Fairness Index close to it.

Proof: From the FVTW algorithm, we can easily see that the time window of a terminal with positive FI will not overlap with the time window of a terminal with negative FI; we can also see that the ending point of the time window of an MT with negative FI is always earlier than the starting point of the time window of an MT with positive FI.

Property 2: For those agents with all negative Fairness Indices, the agent with the smaller FI has higher probability to be chosen as the CHA.

Proof: Assume two MAs u, v that have FIs $FI(u) < FI(v) < 0$. The corresponding time windows of these two agents are: $(A, B - \sum_{i=1}^{|FI(u)|}(\frac{B-A}{2})^i)_u$ and $(A, B - \sum_{i=1}^{|FI(v)|}(\frac{B-A}{2})^i)_v$ respectively. These two time windows have the same starting time point but the time window of agent v is larger than the time window of agent u by $\sum_{i=|FI(u)|+1}^{|FI(v)|}(\frac{B-A}{2})^i$. According to the RTCHS algorithm the timeout point is uniformly distributed in the corresponding time window, therefore agent u which has a smaller time window (but the same starting time point with agent v), has higher probability to be chosen as CHA.

Property 3: For those agents with all positive Fairness Indices, the agent with the smaller FI has higher probability to be chosen as the CHA.

Proof: Assume two agents u, v that have FIs $0 < FI(u) < FI(v)$. The corresponding time windows for these two terminals are: $(A+\sum_{i=1}^{FI(u)}(\frac{B-A}{2})^i, B)_u$ and $(A+\sum_{i=1}^{FI(v)}(\frac{B-A}{2})^i, B)_v$ respectively. These two time windows have the same ending time point but the time window of agent u is larger than that of agent v by $\sum_{i=FI(u)+1}^{FI(v)}(\frac{B-A}{2})^i$. According to the RTCHS algorithm the timeout point is uniformly distributed in the corresponding time window, and therefore agent u which has the larger time window (but the same ending time point with terminal v), has higher probability to be chosen as CHA.

Property 4: In a cluster Φ with Q agents, if all terminals remain in the same cluster during successive cycles of the cooperation process, the Fairness Index of any agent in the cluster is bound within the interval $[1 - Q, Q - 1]$.

Proof: Let us assume that at the beginning all terminals are at their fair status. That is: $FI_i = 0$, $i = 1, 2, 3...Q$. In the following we consider the operation of FVTW-RTCHS scheme in successive CHA selection cycles. Based on Corollary 2, since all terminals start with the same FI they have the same probability to be selected as CHAs at the beginning. Therefore without loss of generality we assume that terminal 1 is selected as the CHA in CHA selection cycle 1. As a result $FI_1 = Q - 1$ and $FI_i = -1$ for all $i > 1$. Because of

property 1, in the next CHA selection cycle (e.g. cycle 2) the CHA will be selected among terminals $i, i = 2, 3, ...Q$. All these terminals have the same FI and therefore the same probability to be selected as CHA. Without loss of generality we assume that terminal 2 is selected as CHA in this cycle. After that operation the values of the FIs of all the terminals are as follows: $FI_i = Q - 2$ for all $i \leq 2$ and $FI_i = -2$ for all $i > 2$. In general, repeating the same process, we can easily conclude that after selection cycle j we have: $FI_i = Q - j$ for all $i \leq j$ and $FI_i = -j$ for all $i > j$. Similarly after selection cycle $(Q-1)$ $FI_i = 1$ for all $i \leq Q-1$ and $FI_i = 1-Q$ for $i = Q$. In the next cycle only terminal Q will be selected as CHA according to FVTW-RTCHS algorithm. At that point $FI_i = 0$ for all nodes. As a result the cooperation process will repeat from the initial status. Therefore from the above cooperation process we conclude that the Fairness Index of any node in the cluster is bound within the interval $[1 - Q, Q - 1]$.

5.5 Performance Evaluation

In this section, we evaluate the performance of the proposed algorithms in terms of achievable energy savings and fairness. In order to quantify and obtain a better understanding of the improvements achieved by the cooperative schemes RTCHS and FVTW-RTCHS, we compare their performance with the corresponding performance of a conventional direct reporting mechanism where each agent individually updates its information to the base station adopting the one-to-one resource management information exchange paradigm. In the following, we first describe the model and assumptions used throughout our performance study. Then we present the corresponding numerical results that demonstrate the achievable energy savings and fairness, and discuss some tradeoffs of the design parameters of the proposed cooperative approach.

5.5.1 Model and Assumptions

In the following we consider a single cell system which is represented by a circle with radius of 1500 meters. All agents at mobile terminals are assumed to update their management information (bandwidth reservation and location) to the base station every 3 seconds. The mobile terminals are randomly distributed in the cell and each terminal is assigned with a random initial position inside the cell, a random moving direction and an initial moving speed which is chosen according to a uniform distribution in the interval [0, 24]meters/second. The speed (v) and direction (ϕ) are updated every 10 seconds according to the following model:

$$v_{new} = \begin{cases} \min\{\max[v_{old} + \Delta v, 0], 24\} & \text{when } p \leq 0.9 \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

$$\phi_{new} = \phi_{old} + \Delta\phi \tag{5.12}$$

where Δv models the acceleration/deceleration of the mobile user and is a uniformly distributed random variable over the interval $[-2meters/second, 2meters/second]$; $\Delta\phi$ characterizes the user's change in moving direction and is a uniformly distributed variable over the interval $[-0.5\pi, 0.5\pi]$; p is a uniformly distributed random variable over the interval $[0, 1]$. The use of variable p allows us to simulate the situation where a mobile user may stop occasionally during the course of moving.

In most of the following experiments and results, unless otherwise indicated, all terminals are assumed to be moving inside the cell according to the model described above, and at each information update time point different terminals may form clusters with different partners. The sizes of the various messages involved in the corresponding experiments are as follows: size of management information update of each terminal is $m_{mgt} = 96$ bits, and size of the query message is $m_{query} = 80$ bits.

5.5.2 Numerical Results and Discussion: Energy Consumption

In figure 5.5, we compare the average total energy consumed by the whole system for the information management updates under the various strategies, as a function of the query range (cluster size) for different values of parameter k, in a 200-terminal wireless network. It should be noted here that the total energy consumed by the cooperative approaches includes both the energy used for the transmission of the actual management information updates to the CHA and from there to the base station, as well as the additional energy required for the creation of the clusters (e.g. cluster formation and query-reply interaction).

As expected the total energy consumption under the direct reporting scheme remains the same, and independent of the query range. RTCHS and FVTW-RTCHS present very similar performance, both outperforming significantly the conventional reporting mechanism in most of the cases. Furthermore as can be seen in figure 5.5 the energy savings achieved by the cooperative approaches increase significantly as parameter k increases.

For both cooperative reporting schemes (RTCHS and FVTW-RTCHS) at the beginning the total energy consumption decreases as the query range r increases, while after some point it starts increasing again with the increase of the query range. As can be observed in figure 5.5 in some special cases when the query range increases a lot (e.g. for the case of $k = 0$ when the range is larger than 500 meters), it is possible that the energy usage of the cooperative schemes may even become larger than the corresponding one under the conventional direct reporting scheme. As we analyzed in section 5.2, if the ratio l_i of the distance between the MEA and the CHA to the distance between the CHA to the base station is $l_i < \sqrt{\frac{24}{5}}$, then on average the whole system saves some energy for the transmission of the management information updates. However as mentioned before some additional energy is consumed by the system in order to form the clusters. The larger the query range (cluster size), the larger the overhead of forming the cluster. Therefore there is some limit on the query range, beyond which the cooperative scheme may even under-perform the conventional direct reporting scheme. Furthermore as can be seen by

figure 5.5 the optimal value of the query range where the energy consumption reaches its minimum value under the cooperative schemes moves to the right - towards larger values of the cluster sizes - as parameter k increases (e.g. from 300 for k=0 to 400 for k=0.5). Similar observation can be made from figure 5.6 where the corresponding results for a 400-terminal scenario are presented.

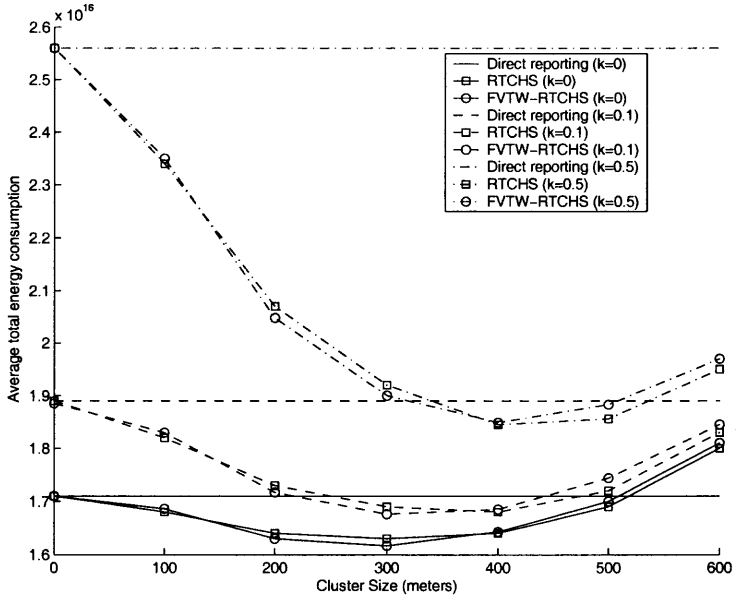


Figure 5.5 Average total energy consumption for 200 terminals vs. query range

5.5.3 Numerical Results and Discussion: Cluster Sizes

As discussed in section 5.2, in cooperative schemes the CHAs may consume much more energy than their member agents, depending on the number of member agents and the corresponding position of the CHA. In order to avoid situations where an agent that is selected as CHA exhausts its energy very fast, we need to limit the cluster size so that only a limited number of member agents are included in one cluster. Figure 5.7 presents the average number of clusters formed at different query ranges when total number of terminals in one cell are 100, 200 and 400 respectively. As can be seen from this figure for the case of 400 terminals when the query range varies from 100 meters to 600 meters the

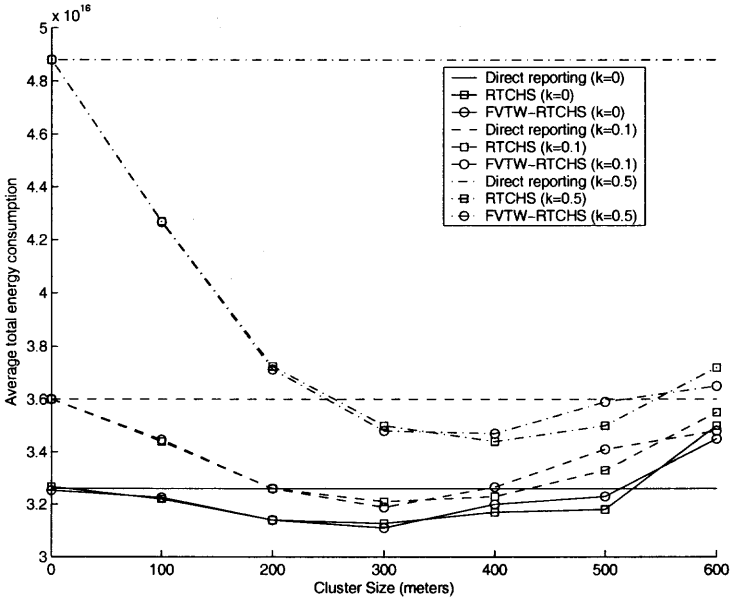


Figure 5.6 Average total energy consumption for 400 terminals vs query range

average number of clusters formed varies from 193 to 18.7. As a result the average number of terminals in one cluster (including the cluster head) may vary from 2.07 to 21.39, which means that at certain time points, a CHA may consume its energy much faster than the normal usage. However at later points, due to the proposed cooperative approach, these nodes will obtain significant energy savings by participating in the clusters of other CHAs and using the two-hop approach to transmit their management information updates to the base station.

5.5.4 Numerical Results and Discussion: Fairness

Figure 5.8 presents the changes of the fairness index of a randomly selected terminal (e.g. terminal 0) as time evolves in successive cooperative reporting operations and CHA selections under the RTCHS and FVTW-RTCHS schemes in the 400-terminal scenario. We can easily see from this figure that without the fairness enhancement (e.g. under the RTCHS scheme), the FI of terminal 0 changes irregularly with very large deviations (from -5 to $+150$) from the fair status (i.e. $FI = 0$). On the other hand under the

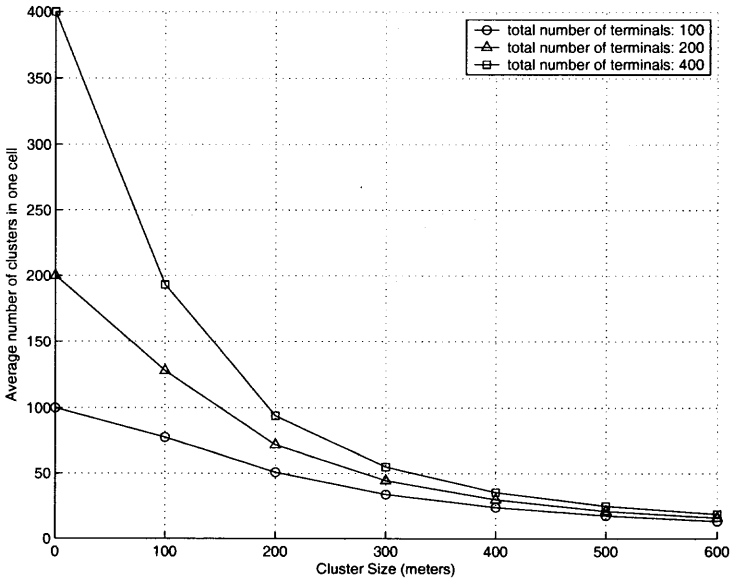


Figure 5.7 Average number of clusters vs query range

FVTW-RTCHS operation the FI of the same terminal 0 has smaller deviations around its fair status (from -10 to $+40$). This happens because the Fair-Variable-Time-Window algorithm acts as a feedback controller to adjust the fairness index from any unfair status (positive or negative) to its fair status. The first and second moments of the FI of terminal 0 for 2000 successive CHA selection cycles under the FVTW-RTCHS scheme are -0.10 and 69.62 respectively which confirms the fairness aspect of the FVTW-RTCHS scheme, while the corresponding values under the RTCHS scheme are 77.72 and 7487.3 respectively. It should be noted however here that although the fairness index (as shown in figure 5.8) under the FVTW-RTCHS scheme achieves smaller deviation, we still can not guarantee that the FI is bounded because terminals from different clusters with different FIs could re-form a new cluster as the system evolves and the nodes move. However, if the members of a cluster keep unchanged during successive cooperative reporting procedures (this could be the case for wireless users that are not moving too fast or too often, such as in wireless LAN networks), the FVTW-RTCHS scheme can guarantee the bound of the FIs as it was

shown in property 4 in section 4.2 and the system can achieve better performance in terms of fairness.

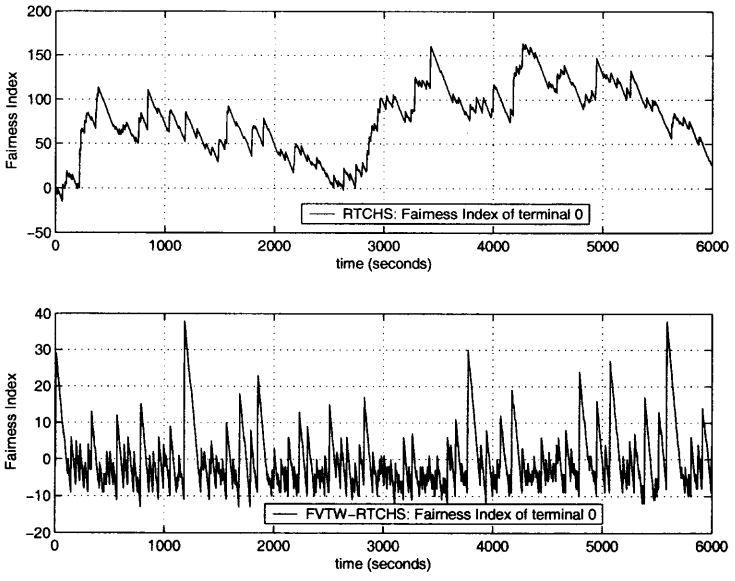


Figure 5.8 Fairness Index evolution vs. time under RTCHS and FVTW-RTCHS

To demonstrate this we have performed a controlled experiment where we constrained 10 terminals within a small area and set the query range appropriately to 300 meters so that these 10 terminals can always form the same clusters, no matter which terminal is selected as the cluster head throughout the successive cooperative reporting processes during the duration of our experiment. We recorded the changes of the FIs for two randomly selected terminals, i.e. terminals 1 and 3, and in figure 5.9 we present the corresponding results under the RTCHS and FVTW-RTCHS algorithms. From this figure we see that under RTCHS operation the FIs of both terminals deviate significantly from their fair status, while under the FVTW-RTCHS operation the corresponding FIs are always bounded within $[-9, +9]$. The actual changes of the FIs for the two mobiles under the FVTW-RTCHS algorithm for the time interval between points 600 and 900 seconds, can be more closely observed in figure 5.10.

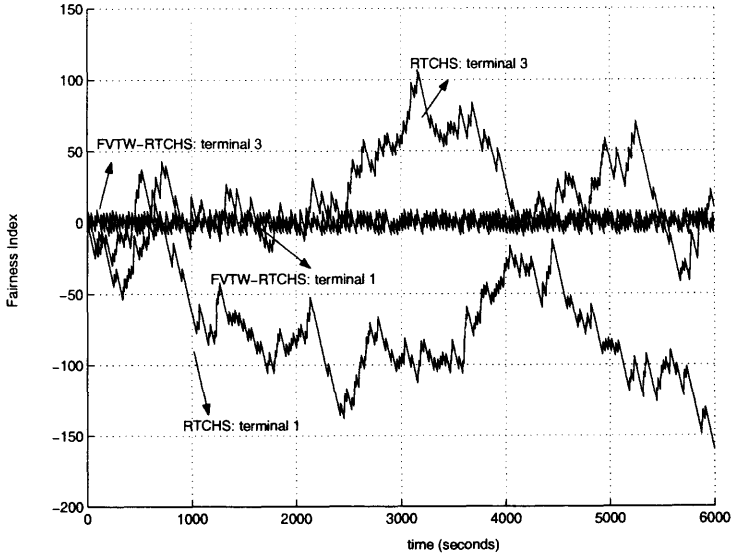


Figure 5.9 FI evolution of the two terminals vs. time under RTCHS and FVTW-RTCHS

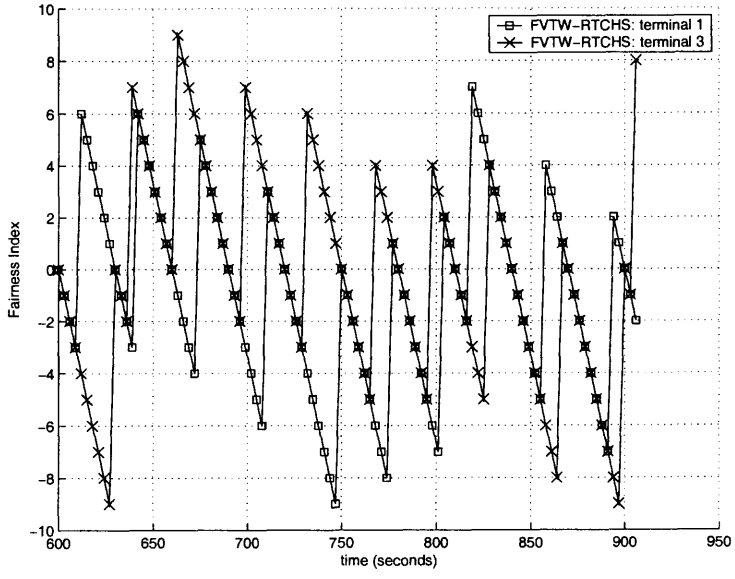


Figure 5.10 Fairness Index of the two terminals vs time under FVTW-RTCHS

5.6 Concluding Remarks

In traditional cellular networks the implementation of management functions, such as bandwidth reservation and location updates, are based on the direct information exchange paradigm between the mobile terminals and the base station, which could prove to be costly in terms of transmit energy consumed, especially in next generation wireless networks where a highly dense network is expected.

In this chapter we designed and demonstrated the use of a distributed cooperative scheme that can be applied in the future wireless networks to improve the energy consumption for the routine management processes of mobile terminals, by adopting the peer-to-peer communication concept of wireless ad-hoc networks. The proposed scheme consists mainly of two parts: the distributed clustering algorithm (Random Timeout Cluster Head Selection - RTCHS) and the query-reply-report information delivery interactions among the cluster head, member agents and the base stations. The performance evaluation process and the corresponding numerical results presented here demonstrated the significant system energy savings that can be achieved through the use of the cooperative approach. Furthermore several tradeoffs of parameters involved in the design of the proposed scheme, such as the cluster size, the query range, etc. were discussed and evaluated.

Another critical issue, associated with the effectiveness of the distributed cooperative scheme is the fairness in the process of selecting the various cluster heads. Based on our discussion the RTCHS algorithm, although provides a fair scheme for every agent in every independent CHA selection cycle, it may introduce some unfairness in successive CHA selection processes (cycles), due to its memoryless characteristic and to the fact that nodes may re-form new clusters as the system evolves and the terminals move. Therefore to improve the fairness of the RTCHS algorithm, a Fair Variable Time Window (FVTW) enhanced RTCHS (FVTW-RTCHS) algorithm is also proposed and evaluated, which uses

the Fairness Index as a feedback from the previous selection cycles to appropriately adjust the probability that an agent is selected as CHA.

It should be noted that in the proposed FVTW-RTCHS algorithm the FI of an agent increases by one credit for each terminal it supports when it acts as CHA, independent of the relative positions of the CHA and the supported agents. However by supporting one member agent in the cluster, CHAs with different positions in the cell may consume different energy. In order to let the fairness index reflect the energy consumption of cluster heads, the credits could be weighted. For example one such approach, that still maintains low complexity, could be to evenly divide the cell in H grids (circular zones), where the CHA in grid h will have its obtained credits weighted by h^n (where n takes values between 2 and 4 due to the wireless transmission energy consumption rule). Correspondingly, agents could get weighted negative credits. The study and design of such enhanced algorithms would be the future research work.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

One of the main features that will strongly characterize the future telecommunications is heterogeneity. Various access technologies and standards co-exist in the world today, and the future communication network that will evolve from the current networks, is expected to be a combination of various multi-domain, multi-provider, multi-technology systems. For instance today's many different wireless systems, ranging from Personal Area Networks (PANs), Wireless Local Area Networks (LANs) to wide-area cellular systems, are often not compatible with each other, which makes it difficult for a user to move from one system to another. In order to meet the objectives of instant adaptability to the users' requirements and of interoperability and seamless operation within the heterogeneous 4G environment, flexibility in terms of network and resource management will be a key design issue for the future network architectures.

The new emerging technology of agent programming has arisen in the distributed programming field as a flexible and complementary way of managing resources of a distributed system, and is a challenging opportunity for delivering more flexible services and dealing with network programmability. Distributing intelligence across the network allows the fast exploitation of more advanced services that can dynamically adapt to the users' requirements. Therefore in this dissertation the mobile agent technology is studied as the underlying vehicle and paradigm for providing flexible and dynamic resource management services. Specifically this dissertation mainly focuses on: a) the design of models that provide a generic framework for the evaluation and analysis of the performance and tradeoffs of the mobile agent management paradigm; b) the development of MA based resource and network management applications. The analytical models developed in this dissertation can be used as a tool to evaluate the candidate various management paradigms for specific tasks and applications. Furthermore, the use of the mobile agent technology

is considered and demonstrated in various network and resource management applications for different networking environments.

For instance, with the rise of packet-oriented wireless networks and powerful mobile terminals, various applications/services are under development such as location-aware navigation guides, financial applications, personal assistance services and entertainment. Future mobile communication systems will be characterized by high throughput, integration of services, heterogeneity, flexibility and adaptability. Although the out-coming new radio technology will bring more bandwidth at air interface, if not managed properly, the bandwidth resources will not meet the requirements of future users. These trends, along with the convergence of communications, information, commerce and computing, are creating a significant demand and opportunity for multimedia personal communication services. Recognizing this trend and need, in this dissertation, we introduced and designed a management framework, based on Mobile Agent technology, that supports, with flexibility and efficiency, the functionality of resource management, and thus can meet the objective of providing flexible QoS-enabled seamless multimedia services to mobile users in next generation mobile communication systems.

Given the observation that over the past twenty years there has been a steady progress towards making computers more mobile, and considering the fundamental energy/power limitations of the mobile terminals and of radio technology, the efficient and fair distributed cooperation among the mobile agents, in order to perform the corresponding resource management functions with energy efficiency, is introduced, investigated and analyzed in this dissertation.

6.1 Contributions

The specific contributions of this dissertation are summarized as follows:

- The use and the benefits of the mobile agent based management paradigm in the network and resource management process is demonstrated, through the introduction

of a commercial application of a multioperator network, and the investigation of the application of agents to provide the underlying framework and structure for its implementation and deployment. It is demonstrated that the mobile agent technology helps overcome several implementation and design issues associated with the need for distributed operation of various network and resource management tasks and algorithms. Such considerations include balancing of the resource management computational load in the network, dynamic fault tolerance, customized service provisioning.

- A general analytical model and framework for the evaluation of various network management paradigms is introduced and discussed. It is also illustrated how the developed analytical framework can be used to quantitatively evaluate the performances and tradeoffs in the various computing paradigms, by comparing the performances of the mobile agent based paradigm with the corresponding ones of the Client-Server mode, under different scenarios. Guidelines and observations about the conditions that the MA based approach outperforms the traditional CS approach are also provided.

- The design tradeoffs for choosing the MA based management paradigm to develop a flexible resource management scheme in wireless networks are discussed and evaluated. The integration of an advanced bandwidth reservation mechanism which facilitates the efficient and seamless operation of the handoff process, with a bandwidth reconfiguration based call admission control strategy that supports flexible QoS management, is also proposed. A framework based on the technology of mobile agents is introduced for the efficient implementation of the proposed integrated resource and QoS management.

- The use of a distributed cooperative scheme among the mobile agents is introduced, which can be applied in the future wireless networks, to improve the energy

consumption for the routine management processes of mobile terminals, by adopting the peer-to-peer communication concept of wireless ad-hoc networks. In traditional cellular networks the implementation of management functions, such as bandwidth reservation and location updates, are based on the direct (one hop) information exchange paradigm between the mobile terminals and the base station, which could prove to be costly in terms of transmit energy consumed, especially in next generation wireless networks where a highly dense network is expected. In the proposed cooperative approach the network is subdivided into ad hoc subnetworks (clusters) where the agents of each cluster cooperate in order to perform the required management functions, reducing the overall wireless communication cost. In this kind of cooperative reporting scheme, the conventional individual direct report transmissions of the MAs to the base station are replaced by two-hop transmissions. The performance evaluation process and the corresponding numerical results demonstrate the significant system energy savings that can be achieved through the use of the mobile agent cooperative approach, while several design issues and tradeoffs involved in the proposed scheme, such as the fairness of the mobile agents involved in the management activity, are identified and analyzed.

6.2 Future Work

The main research efforts in this dissertation, focused on the design and analysis of enhanced network and resource management applications based on the mobile agent technology. This study, analyzed several aspects and tradeoffs of the use of mobile agent technology in wired and wireless networks, and demonstrated the flexibility and advantages that it brings into the overall network and resource management process. However, in order to optimize the overall performance of the network management process and the mobile agent system itself, mobile agent behavior oriented models should be also studied and developed. The analytical model developed in this dissertation, mainly considered the

scenario where a mobile agent is serving a group of nodes belonging to one management task. Future work should be dedicated to using this model as a tool to optimize the whole system, which may contain several management sub tasks.

Furthermore, the development and use of different agent dissemination strategies may significantly impact the performance, especially in completing periodical management tasks. Towards that direction, different MA dissemination strategies need to be designed and analyzed. Such strategies may range from the "no-clone" strategy, where the entity of the MA itself with some computational status move to the next node, to the "cloned" strategy, where a copy of the MA is generated and moves to the next node, while the original MA stays at the previous node. The no-clone strategy can keep the number of MAs in the system low while at the same time the memory usage at each node is low as well. The tradeoff is that the traffic of moving the MA among the nodes increases. On the other hand, the clone strategy decreases the traffic and time of transportation of MAs among the nodes when the task is repeated periodically, however the number of MAs co-existing in the system increases, and the memory usage at each node increases as well. More generally, the issue of how two or more management tasks could efficiently share one or more groups of mobile agents during the management task process, in order to minimize the system resource usage overhead, is of research and practical importance.

BIBLIOGRAPHY

[1] G. Goldszmidt and Y. Yemini, "Delegated agents for network management," *IEEE Communications Magazine*, vol. 36, pp. 66–70, March 1998.

[2] A. Sethi, D. Zhu, V. Hnatyshin, and P. Kokati, "Battlefield network applications of the SHAMAN management system," in *Military Communications Conference, MILCOM 2001*, pp. 933–937.

[3] A. Iwata and N. Fujita, "A hierarchical multilayer QoS routing system with dynamic SLA management," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 2603–2616, Dec. 2000.

[4] E. Duarte and T. Nanya, "A hierarchical adaptive distributed system-level diagnosis algorithm," *IEEE Transactions on Computers*, vol. 47, pp. 34–45, Jan. 1998.

[5] R. Kawamura and R. Stadler, "Active distributed management for IP networks," *IEEE Communication Magazine*, vol. 38, pp. 114–120, April 2000.

[6] M. Breugst and T. Magedanz, "Mobile agents - enabling technology for active intellligent network implementation," *IEEE Network*, vol. 12, pp. 53–60, May/June 1998.

[7] V. A. Pham and A. Karmouch, "Mobile software agents: An overview," *IEEE Communications Magazine*, vol. 36, pp. 26–37, July 1998.

[8] D. B. Lange and M. Oshima, "Seven good reasons for mobile agents," *Comminications of the ACM*, vol. 42, pp. 88–89, March 1999.

[9] G. Cabri, L. Leonardi, and F. Zambonelli, "MARS: a programmable coordination architecture for mobile agents," *Internet Computing, IEEE*, vol. 4, pp. 26–35, Jul/Aug 2000.

[10] A. Puliafito, O. Tomarchio, and L. Vita, "Map: Design and implementation of a mobile agent platform," *J. Syst. Architecture*, vol. 46, pp. 256–267, Jan. 2000.

[11] D. Kotz, R. Gray, S. Nog, D. Rus, S. Chawla, and G. Cybenko, "Agent TCL: targeting the needs of mobile computers," *IEEE Internet Computing*, vol. 1, pp. 58–67, Jul/Aug 1997.

[12] A. Ohsuga, Y. Nagai, Y. Irie, M. Hattori, and S. Honiden, "Plangent: an approach to making mobile agents intelligent," *IEEE Internet Computing*, vol. 1, pp. 50–57, Jul/Aug 1997.

[13] D. Putzolu, S. Bakshi, S. Yadav, and R. Yavatkar, "The Phoenix framework: a practical architecture for programmable networks," *IEEE Communications Magazine*, vol. 38, pp. 160–165, Mar. 2000.

[14] M. Greenberg, J. Byington, and T. Holding, "Mobile agents and security," *IEEE Communications Magazine*, vol. 7, pp. 76–85, July 1998.

[15] K. Schelderup and J. Olnes, "Mobile agent security: Issues and directions," in *IS&N'99*, vol. LNCS1597, March 1999.

[16] W. Jansen, "Countermeasures for mobile agent security," *Computer Communications*, vol. 23, pp. 1667–1676, 2000.

[17] I. Iida, T. Nishigaya, and K. Murakami, "Duet: An agent-based personal communication network," *IEEE Communication Magazine*, vol. 33, pp. 44–49, November 1995.

[18] M.-P. Gervais and A. Diagne, "Enhancing telecommunications service engineering with mobile agent technology and formal methods," *IEEE Communications Magazine*, vol. 36, pp. 38–43, Jul. 1998.

[19] S. Gonzlez-Valenzuela and V. Leung, "QoS routing for mpls networks employing mobile agents," *IEEE Network*, vol. 16, pp. 16–21, May/Jun 2002.

[20] M. Gunter and T. Braun, "Internet service monitoring with mobile agents," *IEEE Network*, vol. 16, pp. 22–29, May/Jun 2002.

[21] H. D. Meer, A. L. Corte, A. Puliafito, and O. Tomarchio, "Programmable agents for flexible QoS management in IP networks," *IEEE Journal on Selected Areas in Communication*, vol. 18, pp. 145–162, February 2000.

[22] Y. Wijata, D. Niehaus, and V. Frost, "A scalable agent-based network measurement infrastructure," *IEEE Communications Magazine*, vol. 38, pp. 174–183, Sep 2000.

[23] D. Kotz, G. Cybenko, R. Gray, G. Jiang, and R. Peterson, "Perfomance analysis of mobile agents for filtering data streams on wireless networks," *Mobile Networks and Applications*, vol. 7, pp. 163–174, April 2002.

[24] S. H. Kim and T. G. Robertazzi, "Mobile agent modeling," in *CISS 2002*, March 2002.

[25] M. Baldi and G. Picco, "Evaluating the tradeoffs of mobile code design paradigms in network management applications," in *Proceedings of Conference on Software Engineering*, pp. 146–155, April 1998.

[26] A. Fuggetta, G. P. Picco, and G. Vigna, "Understanding code mobility," *IEEE Transactions on Software Engineering*, vol. 24, pp. 342–361, May 1998.

[27] T. M. Chen and S. S. Liu, "A model and evaluation of distributed network management approaches," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 850–857, May 2002.

[28] Z. Wang and J. Crowcroft, "Quality-of-Service routing for supporting multimedia applications," *EEE JSAC*, pp. 1228–1234, September 1996.

[29] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*. Springer-Verlag, 1992.

[30] K. Uchimura and H. Sakaguchi, "Vehicle routing problem using genetic algorithms based on adjacency relations," in *Vehicle Navigation and Information Systems Conference*, pp. 214–217, 1995.

[31] D. E. Goldberg, *Genetic algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Pub. Co., 1989.

[32] S. Tanterdtid, W. S., and W. B., "Optimum virtual paths system based in ATM network using genetic algorithm," in *ICICS'97*, pp. 596–601, 1997.

[33] S. Papavassiliou, A. Puliafito, O. Tomarchio, and J. Ye, "Mobile agent-based approach for efficient network management and resource allocation: framework and applications," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 858–872, May 2002.

[34] J. Ye, J. Hou, and S. Papavassiliou, "A comprehensive resource management framework for next generation wireless networks," *IEEE Trans. on Mobile Computing*, vol. 1, pp. 249–264, Oct.-Dec. 2002.

[35] Y. Xiao, C. L. Chen, and Y. Wang, "An optimal distributed call admission control for adaptive multimedia in wireless/mobile networks," *Proceedings of 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 477–482, 2000.

[36] T. Kwon, Y. Choi, C. Bisdikian, and M. Naghshineh, "Qos provisioning in wireless/mobile multimedia networks using an adaptive framework," *Wireless Networks*, vol. 9, no. 1, pp. 51–59, 2003.

[37] C. Lee, J. Lehoczky, D. Siewiorek, R. Rajkumar, and J. hansen, "A scalable solution to the multi-resource QoS problem," in *Proc. of the 20th IEEE Real-Time Systems Symposium*, December 1999.

[38] C. Courcoubetis and V. Siris, "Managing and pricing service level agreements for differentiated services," in *Proc. of 7th ITTT/IFIP International workshop on Quality of Service (IWQoS'99)*, June 1999.

[39] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Select. Areas Comm.*, pp. 968–981, Sept. 1991.

[40] F. P. Kelly, "Effective bandwidths at multi-class queues," *Queueing Syst.*, vol. 9, pp. 5–16, 1991.

[41] A. I. Elwalid and D. Mitra, "Effective bandwidth of general markovian traffic sources and admission control of high-speed networks," *IEEE/ACM Trans. Networking*, pp. 329–343, 1993.

[42] J. Hou and S. Papavassiliou, "Influence-based channel reservation scheme for mobile cellular networks," in *Proceedings of IEEE ISCC 2001*, pp. 218–223, July 2001.

[43] P. Brockwell and R. Davis, *Time Series: Thoery and Methods, 2nd ed.* New York: Springer Verlag, 1991.

[44] M. Chiu and M. A. Bassiouni, "Predictive schemes for handoff prioritization in cellular networks based on mobile positioning," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 510–522, March 2000.

[45] D. A. Levine, I. F. Akylidiz, and M. Naghshineh, "A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept," *IEEE/ACM Transactions on Networking*, vol. 5, February 1997.

[46] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures," *IEEE Trans. on Vehicular Technology*, vol. 35, pp. 77–92, August 1986.

[47] C. Oliveira, J. B. Kim, and T. Suda, "An adaptive bandwidth reservation scheme for high-speed multimedia wireless network," *IEEE Journal on Selected Area in Communications*, vol. 6, pp. 858–874, August 1998.

[48] O. Kachirski and R. Guha, "Intrusion detection using mobile agents in ad hoc wireless networks," *2002 Proceedings of IEEE Workshop on knowledge Meadia Networking*, 2002.

[49] F.-C. Lin and C.-N. Kuo, "Cooperative multi-agent negotiation for electronic commerce based on mobile agents," *2002 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, p. 6, Oct. 2002.

[50] F. Morvan, M. Hussein, and A. Hameurlain, "Mobile agent cooperation methods for large scale distributed dynamic query optimization," *In Proceedings of 14th International Workshop on Database and Expert Systems Applications, 2003*, pp. 542–547, 2003.

[51] K.-D. Lin and J.-F. Chang, "Communications and entertainment onboard a high-speed public transport system," *IEEE Wireless Communications*, vol. 9, pp. 84–89, Feb. 2002.

[52] N. Davies, K. Cheverst, A. Friday, and K. Mitchell, "Future wireless applications for a networked city: services for visitors and residents," *IEEE Wireless Communications*, vol. 9, pp. 8–16, Feb. 2002.

[53] J. Krikke, "Graphics applications over the wireless web: Japan sets the pace," *IEEE Computer Graphics and Applications*, vol. 21, pp. 9–15, May/June 2001.

[54] A. Smailagic, D. Siewiorek, and M. Ettus, "System design of low-energy wearable computers with wireless networking," in *Proceedings of IEEE Computer Society workshop on VLSI, 2001*, pp. 25–29, May 2001.

[55] A. Scaglione and Y.-W. Hong, "Opportunistic large arrays: Cooperative transmission in wireless multihop Ad Hoc networks to reach far distance," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2082–2092, August 2003.

[56] S. K. Shah, S. Tekinay, and C. Saraydar, "Co-operative location updating for mobiles in next generation cellular networks," *Proc. IEEE High Performance Switching and Routing 2004*, April 2004.

[57] H. Li, M. Lott, M. Weckerle, W. Zirwas, and E. Schulz, "Multihop communications in future mobile radio networks," *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, 2002.

[58] C.-T. Chen, S. Tekinay, and S. Papavassiliou, "Geocasting in cellular ad hoc augmented networks," *Proc. IEEE VTC 2003-Fall*, October 2003.

[59] Y.-D. Lin and Y.-C. Hsu, "Multihop cellular: a new architecture for wireless communications," *INFOCOM 2000*, vol. 3, pp. 1273 – 1282, March 2000.

[60] H. Wu, C. Qiao, S. De, and O. Tonguz, "Integrated cellular and ad hoc relaying systems: icar," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 2105–2115, Oct. 2001.

[61] A. Kusuma and L. Andrew, "Minimum power routing for multihop cellular networks," *GLOBECOM '02*, vol. 1, pp. 37 – 41, Nov. 2002.

[62] T. Rappaport, *Wireless Communications principles and Practice*. Prentice-Hall, 1996.

[63] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1265–1275, Sept. 1997.

[64] J.-H. Ryu, S. Song, and D.-H. Cho, "Energy-conserving clustering scheme for multicasting in two-tier mobile ad-hoc networks," *IEEE Electronics Letters*, vol. 37, pp. 1253–1255, Sept. 2001.

[65] Z. Cai, M. Lu, and X. Wang, "Channel access-based self-organized clustering in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 102–113, April-June 2003.